SUBDIVISION SURFACES IN MPEG-4

Francisco Morán[1], Patrick Gioia[2], Michael Steliaros[3], Mikaël Bourges-Sévenier[4] and Narciso García[5]

[1,5]Grupo Tratamiento Imágenes
Univ. Politécnica Madrid, Spain
{fmb,ngs}@gti.ssr.upm.es

[2]France Télécom R&D
Patrick.Gioia@rd.francetelecom.com

[3]Superscape (UK), Ltd.
MSteliaros@superscape.com

[4]Mindego, Inc.
mikael@mindego.com

ABSTRACT

SSs (Subdivision Surfaces) are a powerful modelling paradigm for *truly hierarchical* (instead of *merely progressive*) 3D *surface* (as opposed to *mesh*) coding. Two SS-based tools of MPEG-4's Animation Framework eXtension allow to derive a piecewise smooth surface from an initial control mesh: if the subdivision process is run in its predefined form, the initial mesh is simply smoothed; if 3D details are added to the positions of the new vertices appearing after each subdivision step, a particular target surface may be approximated with an increasing accuracy. In both cases, multiresolution editing/animation is possible.

1. INTRODUCTION

1.1. Traditional surface tilings *vs.* subdivision surfaces

The MPEG-4 standard already included in its version 2 [10] tools for the efficient coding of 3D surfaces, but they were based in the simplest approximation of a surface: the one resulting from tiling it with planar facets — in practice, with triangles, since they are the simplest polygons. The problem with such a linear approximation to an arbitrarily complex surface is that hundreds of thousands of elements (vertices, edges, facets) are easily needed to obtain a reasonable approximation accuracy. Moreover, the editing or animation of a polygonal mesh is cumbersome, because its vertices are semantically unrelated, and must therefore be moved individually.

This is why most CAD (Computer-Aided Design) and 3D modelling commercial applications still use curved patches of the NURBS (Non-Uniform, Rational, B-Spline) family for tiling 3D surfaces. Patches provide a compact, convenient method to generate piecewise smooth, higher-order surfaces from relatively few control points, whose movement deforms locally the surface. However, patch control grids must be perfectly regular, so modelling objects of arbitrary topology with NURBS introduces non-trivial patch-stitching and curve-trimming difficulties. Furthermore, when modelling intricate surfaces, a large number of tiles (either curved or planar) are needed to describe high-frequency regions, so patches may not provide a much more compact solution than polygons.

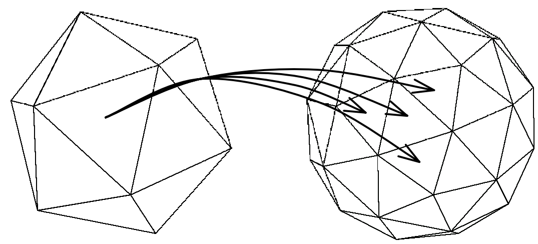

Figure 1: Splitting an icosahedron to get a sphere (1^{st} step).

SSs (Subdivision Surfaces) [14][9] establish a bridge between polygons and patches, as subdivision schemes define simple and efficient mechanisms to derive a smooth surface from an initial control polyhedron of arbitrary topology. Indeed, SSs are defined as the limit of a refinement process of both the connectivity and the geometry of a planar mesh which recursively splits each of its elements (usually its facets) into several ones (usually four: see Figure 1). If this process is carefully designed, the mesh tends to a limit surface as smooth as NURBS-based ones.

Subdivision is extremely useful for approximating and manipulating a surface at different LODs (Levels Of Detail). The successive control meshes, usually called LODs themselves, are pyramidally nested, and define inherently a multiresolution model of the limit 3D surface, as the vertex positions of different LODs are hierarchically related: those of vertices of LOD n (*i.e.*, appearing at the n^{th} subdivision step) are determined by those of some set of neighbour vertices of LOD $n-1$, …, in turn determined by those of the initial (LOD 0) vertices. It is thus easy to perform large-scale edits, in which the movement of a few vertices of a coarse control mesh drags a wide area of the surface, as well as minute detail modifications, in which only a few vertices of the finest meshes are displaced.

1.2. Subdivision surfaces in MPEG-4

SSs in the AFX (Animation Framework eXtension) [11] toolset, to be released in version 5 of the MPEG-4 standard at the end of 2002, come in two flavours, depending on whether the positions of the new vertices appearing after each subdivision step may be modified or not before splitting the mesh again. In both cases, a piecewise smooth limit surface is obtained by subdividing the initial control mesh. But in the "basic SSs setting" (described in section

2), that limit surface is completely defined by the initial control mesh, which is simply smoothed, whereas in the "detailed/wavelet SSs scenario" (see section 3), the aim is to approximate a particular target surface with an increasing accuracy thanks to the 3D details added to the predicted new vertices at each step of the subdivision process.

2. BASIC SUBDIVISION SURFACES

2.1. Traditional subdivision schemes

One of the main differences between subdivision schemes resides in the kind of polygons they act upon: the (non-smoothing) midpoint scheme operates on triangles only, as do Loop's [6] and Dyn's "butterfly" [3]; instead, Catmull-Clark's [2] accepts generic polyhedra, which are turned into quadrilateral meshes in a pre-processing step. Another key difference between subdivision schemes is their interpolating *vs.* approximating character: in the case of interpolating schemes, like the midpoint or butterfly ones, the limit surface interpolates all vertices of all control meshes; for approximating ones, like Loop's or Catmull-Clark's, control points need not lie on the limit surface.

2.2. Normal control and tagging

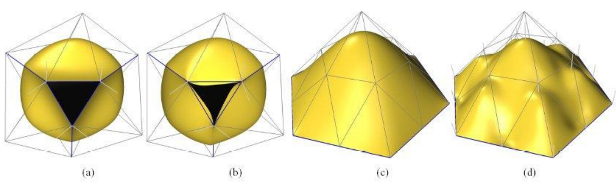

Figure 2: Normal control: (a) surface with convex corners; (b) Prescribed normals at each corner. (c) Smooth surface. (d) Same control mesh but all normals vertical.

The original schemes of Loop and Catmull-Clark suffer from some problems addressed by Biermann [1]: by introducing sector tagging, correct treatment of concave/convex corners, flatness modification within sectors, and boundary and interior normal control are possible (see Figure 2).

All schemes allow edge and vertex tagging. Edges may be tagged as "creased" to prevent locally the control mesh smoothing (for instance, boundary edges are automatically assigned crease tags). Moreover, vertices at either side of a crease edge may be tagged as "creased" (to join exactly two incident crease edges smoothly), "corner" (to join two or more crease edges in a corner), or "dart" (to have the crease edge blend smoothly into the surface). If no vertex tags are specified, the default tags for vertices with one, two, and three or more crease edges in their neighborhood are dart, crease and corner respectively.

2.3. Extended Loop scheme

The extended Loop scheme is an enhancement of Loop's that allows the designer to work with quadrilateral-based models, which are better suited to capturing the symmetries of natural and man-made objects, while ensuring that the visualisation process can be optimised for a triangle-only rendering pipeline. Furthermore, unlike quadrilateral-based subdivision schemes such as Catmull-Clark's, extended Loop offers the designer exact control over the final model triangulation, which is essential for efficient rendering of low polygon models.

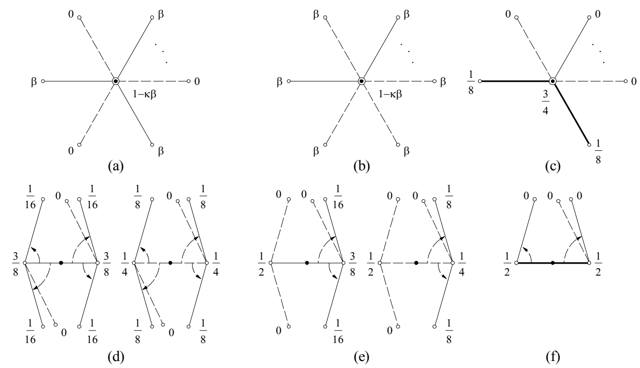

Figure 3: Extended Loop subdivision stencils for interior smooth (a), interior dart (b), boundary/crease vertices (c) and splitting interior (d), and boundary/crease edges (e).

Figure 3 shows the extended Loop subdivision stencils used for repositioning old vertices (as with standard Loop, corner vertices are never moved) and splitting edges. Solid lines indicate original edges while dashed lines indicate "invisible" edges (a tag used to keep track of edges produced by the triangulation pre-processing step). Stencils (a), (b) and (c) show that repositioning old vertices is identical to standard Loop if invisible edges are ignored, with one exception, stencil (b): if a vertex has two or less visible edges in its neighborhood, then the invisible edges are included in the calculation. As with standard Loop,

$$\beta = \frac{1}{k}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2\right),$$

where k is the number of visible edges (Figure 3a) or the valence (visible + invisible edges) if the number of visible edges is strictly less than three (Figure 3b). The stencil for splitting boundary/crease edges (Figure 3f) is identical to standard Loop, while splitting interior edges depends on whether the edge in question is visible (Figure 3d-left) or not (Figure 3d-right). If no visible edges exist for an endpoint then a full weight is used (Figure 3e).

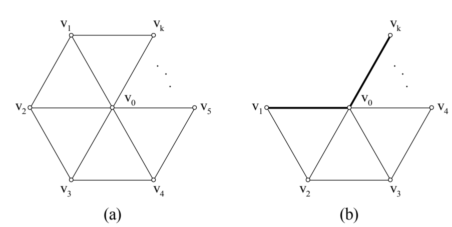

Figure 4: (a) Interior, and (b) boundary/crease vertex neighborhood used for normal calculation.

In order to perform operations such as lighting on the subdivision surface, two tangent vectors are defined for each type of vertex (interior, smooth crease, corner, dart).

For interior smooth and dart vertices of valence k (Figure 4a), the tangents are defined as the weighted averages of the positions of the neighbour vertices:

$$t_1 = c_1 v_1 + c_2 v_2 + c_3 v_3 + \ldots + c_k v_k$$
$$t_2 = c_2 v_1 + c_3 v_2 + c_4 v_3 + \ldots + c_k v_{k-1} + c_1 v_k ,$$

where $c_i = \cos\frac{2\pi i}{k}$. For boundary, crease and corner vertices (Figure 4b), each sector between pairs of boundary/crease edges will have a different pair of tangent vectors that are defined as:

$$t_1 = v_1 - v_k$$
$$t_2 = \begin{cases} -2v_0 + v_1 + v_2, & k = 2 \\ v_2 - v_0, & k = 3 \\ -2v_0 - v_1 + 2v_2 + 2v_3 - v_4, & k = 4 \\ w_1 v_1 + w_2 v_2 + w_3 v_3 + \ldots + w_k v_k, & k > 4 , \end{cases}$$

where, for $\theta = \frac{\pi}{k-1}$, w_i is defined as:

$$w_i = \begin{cases} \sin\theta, & i = 1, i = k \\ (2\cos\theta - 2)\sin((i-1)\theta), & 2 \le i \le k-1 . \end{cases}$$

2.4. AFX's SubdivisionSurface node

The SubdivisionSurface node allows the specification of a tagged control mesh along with any associated texture coordinate and color information. The subdivision scheme can be Loop's or Catmull-Clark's, with optional sector tagging and normal control specified through an array of SubdivSurfaceSector nodes, or extended Loop.

3. WAVELET SUBDIVISION SURFACES

3.1. Progressive mesh *vs.* hierarchical surface coding

Progressive meshes, as first proposed by Hoppe [5] and later improved by teams such as Taubin's (whose work was the basis for MPEG-4 version 2), provide an efficient means to code a fine polygonal mesh M as a set of LODs in a lossless way, at least in what concerns M's connectivity (the coding of M's geometry is always lossy due to the quantisation of vertex coordinates, although devoting 16 bits to each of $\{x, y, z\}$ is enough to resolve 1 mm details in a model of a building). But if M is only a piecewise linear approximation of a (most likely piecewise smooth) given target surface, it makes no sense to insist on coding its connectivity losslessly, because no particular mesh is better than any of infinitely many other approximating the true surface within a certain tolerance.

In this sense, merely progressive coding of 3D meshes is inferior to truly hierarchical coding of 3D surfaces. The differential refinements that permit to go from one LOD to the next, in a progressive mesh, are randomly located, and there is no relationship between the elements of one LOD and its immediate neighbours. In the case of the LOD sets obtained with SSs, two consecutive LODs are also differentially related, but there is a very clear pyramidal nesting of their elements, which has many practical advantages.

In particular, such a pyramid of LODs is highly suitable for a hierarchical coding (and, hence, transmission) of 3D models thanks to MRA (MultiResolution Analysis) techniques. The main idea behind MRA is to decompose a signal into a coarse, low resolution/frequency part, plus a collection of finer and finer details only observable at increasing resolutions/frequencies. Wavelet-based MRA has been successfully applied in the last decade to classic nD signals ($n = 1 \Rightarrow$ audio; $n = 2 \Rightarrow$ still image; $n = 3 \Rightarrow$ video) [13], but only recently extended to decompose signals defined over 2D manifolds (proper surfaces) [8].

Especially for signal transmission purposes, sub-band decomposition is a very important concept. Not only does it permit to send a coarse version of the signal first and progressively refine it afterwards, but it also enables a more compact coding of the information carried by signals whose energy is mostly concentrated in their low frequency part (in the case of surfaces: mostly smooth ones).

3.2. Subdivision regarded as a prediction mechanism

In a multiresolution predictive coding scheme, a successively approximated signal is decomposed into its coarsest version plus a set of prediction errors, which are the differences between the original signal and the estimates based on its successively refined versions. The predictor is usually designed to generate small prediction errors for mostly smooth signals, like those encountered in practice, and is thus able to transform a possibly energetic signal into a low variance set of prediction errors. Entropy coding techniques can be further applied to the prediction errors, to represent the same amount of information with less data.

Subdivision can be seen as a multiresolution predictive coding scheme, as the rules of a particular scheme are indeed a prediction mechanism, in which the position of new vertices are estimated with those of old ones. Suppose a base control mesh M^0 has been extracted from a given target surface T, which has to be approximated by a set of LODs constructed by subdividing M^0. Suppose also, for the sake of simplicity, that all vertices of M^0 lie on M and that an interpolating subdivision scheme is used. One cannot expect the new vertices to lie precisely on T, but nothing prevents one from correcting their positions so that they do, after the standard subdivision rules have misplaced them, and to use that modified mesh as the input one for the next step of the subdivision process.

A hierarchical 3D model transmission scenario is then obvious: after having transmitted M^0, only the 3D details to be added to the positions of the new vertices need to be sent, if both encoder and decoder have previously agreed upon a set of subdivision rules and hierarchy traversal order. Those details can be seen as prediction errors, because they measure the difference between the predicted vertices and the desired ones lying on T.

3.3. Wavelet coding of 3D details

Consider a triangular scheme, and let M^n be the n-th canonical quadrisection of a base mesh M^0 and $\mathbf{V}^n$ the matrix whose i-th row holds the x, y and z coordinates of the i-th vertex of M^n. It can then be written that the limit surface is $S = \lim_{n\to\infty} M^n$ and that $\mathbf{V}^{n+1} = \mathbf{P}^n\, \mathbf{V}^n$, where $\mathbf{P}^n$ is the global subdivision matrix of the considered scheme for a given base mesh connectivity and subdivision level n [8]. As interesting subdivision schemes are local, the position of any vertex of M^{n+1} depends on those of only a few vertices of M^n, which means that $\mathbf{P}^n$ is sparse and that a small portion of it suffices to characterise the scheme: the local subdivision matrix $\mathbf{P}$, which depends on the connectivity, but not on the level.

Following Lounsbery's extension of classic MRA, it is possible to express $\mathbf{V}^{n+1}$ in terms of $\mathbf{V}^n$ and $\mathbf{D}^n$, a set of wavelet coefficients corresponding to the details above, as:

$$\mathbf{V}^{n+1} = (\mathbf{P}^n\ \mathbf{Q}^n)\begin{pmatrix}\mathbf{V}^n\\ \mathbf{D}^n\end{pmatrix},$$

where $(\mathbf{P}^n\ \mathbf{Q}^n)$ is an invertible matrix formed by $\mathbf{P}^n$, the prediction operator, and $\mathbf{Q}^n$, which interprets the wavelet coefficients. $\mathbf{P}^n$ and $\mathbf{Q}^n$ are called the synthesis filters because they can be seen respectively as the low-pass and high-pass reconstruction filters to be applied to $(\mathbf{V}^n\ \mathbf{D}^n)^{\mathrm{T}}$ to generate $\mathbf{V}^{n+1}$. As $(\mathbf{P}^n\ \mathbf{Q}^n)$ is invertible, its inverse $(\mathbf{A}^n\ \mathbf{B}^n)^{\mathrm{T}}$, a matrix formed by the analysis filters, can be used to extract from any semi-regular triangle mesh of the same base connectivity as M a set of wavelet coefficients $\mathbf{D}^n$.

Regarding computational efficiency, it is important to note that both synthesis matrices, used during decoding, are usually sparse, and can thus be applied in linear time, but that only interpolating schemes yield sparse analysis matrices, so linear encoding time is not achievable with approximating schemes. As for compression efficiency, it must be noted that Said's SPIHT method [12] for zerotree-like entropy coding can be used to exploit magnitude correlation in the details set and significantly reduce its redundancy, once a detail hierarchy is defined [7][9].

3.4. AFX's WaveletSubdivisionSurface node

The WaveletSubdivisionSurface node allows the transmission of such a compact and hierarchical representation of a 3D surface by encapsulating a base mesh together with a completely embedded bitstream containing the SPIHT-like coded wavelet coefficients of the 3D details. Once the base mesh is received, a few flag bits are read which indicate what subdivision scheme/predictor (Loop, extended Loop, butterfly or midpoint) was used to calculate the details and whether they were expressed in one global or many local coordinate frame(s). Then the details bitstream is decoded using the SPIHT algorithm to recover the wavelet coefficients, with which the model is completely or partially rebuilt, according to the terminal processing power.

One of the main features of this node is that the way the bitstream is understood enables complete adaptivity not only in the reconstruction but also in the transmission [4]. Indeed, there are three kinds of adaptive mesh refinements:

1. **View-dependency**: wavelets are well located in space, thus allowing a geometric sorting of the coefficients according to the visibility of the portion of the mesh they refine.
2. **Sub-band decomposition**: partial reconstruction can be achieved by using a subset of the whole wavelet coefficients set, instead of all of it.
3. **Bitstream embedding**: the SPIHT algorithm provides by nature bitplane refinements of wavelet coefficients.

The structure of the bitstream is designed to enable such a selective transmission by reading bitplanes one by one, together with the information saying which part of the base mesh is refined and at which precision.

4. REFERENCES

[1] H. Biermann., A. Levin. and D. Zorin: "Piecewise-smooth subdivision surfaces", *ACM SIGGRAPH'00 proc., 113-120,* July 2000.

[2] E. Catmull and J. Clark: "Recursively generated B-spline surfaces on arbitrary topological meshes", *Computer-Aided Design,* ***10****-6, 350-355*, November 1978.

[3] N. Dyn, D. Levin and J. Gregory: "A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control", *ACM Tr. on Graphics,* ***9****-2, 160-169*, April 1990.

[4] P. Gioia: "Modélisation, codage et visualisation de surfaces maillées par ondelettes pour la transmission progressive et la navigation adaptative", *Ph.D.* thesis, *Université de Rennes 1*, May 2001.

[5] H. Hoppe: "Progressive Meshes", *ACM SIGGRAPH'96 proc., 99-108*, August 1996.

[6] C. Loop: "Smooth Subdivision Surfaces Based on Triangles", Master thesis, *University of Utah*, August 1987.

[7] A. Khodakovsky, P. Schröder and W. Sweldens: "Progressive Geometry Compression", *ACM SIGGRAPH'00 proc., 271-278*, July 2000.

[8] M. Lounsbery, T. DeRose and J. Warren: "Multiresolution Analysis for Surfaces of Arbitrary Topological Type", *ACM Tr. on Graphics,* ***16****-1, 34-73*, January 1997.

[9] Francisco Morán: "Hierarchical modelling of 3D objects with subdivision surfaces", *Ph.D.* thesis, *Universidad Politécnica de Madrid*, October 2001.

[10] MPEG (Moving Picture Experts Group, formally "ISO/IEC JTC1/SC29/WG11"): "ISO/IEC 14496/Amd 1 Extensions" (a.k.a. "MPEG-4 version 2"), *ISO/IEC*, December 1999.

[11] MPEG (main editor: M. Bourges-Sévenier): "Study of ISO/IEC 14496-1:2001/PDAM4: Animation Framework eXtension and Multi-User Worlds" (MPEG doc. N4852), *ISO/IEC*, May 2002.

[12] A. Said and W. Pearlman: "A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", *IEEE Tr. on Circuits and Systems for Video Technology,* ***6****-3, 243-250*, June 1996.

[13] M. Vetterli and J. Kovačević: "Wavelets and Subband Coding", *Prentice-Hall*, 1995.

[14] D. Zorin: "Subdivision and Multiresolution Surface Representations", *Ph.D.* thesis, *CalTech*, September 1997.