

REAL-TIME RECONSTRUCTION OF WAVELET ENCODED MESHES FOR VIEW-DEPENDENT TRANSMISSION AND VISUALIZATION

Patrick Gioia, Olivier Aubault, Christian Bouville

France Telecom R & D

E-mail: {patrick.gioia, olivier.aubault, christian.bouville}@rd.francetelecom.com

ABSTRACT

Wavelet methods for geometry encoding is a recently emerged superset of multiresolution analysis which has proven to be very efficient in term of compression and adaptive transmission of 3D content. The decorrelating power and space/scale localization of wavelets enable efficient compression of arbitrary meshes as well as progressive and local reconstruction. Recent techniques based on zerotree compression have shown to be among the best lossy mesh compression methods, while remaining compatible with selective transmission of geometric data at various level of detail. While some progressive reconstruction schemes have been proposed in the past, we show in this paper that this representation, recently proposed in the MPEG4 standard, can be efficiently used to perform real-time, view-dependent reconstruction of large meshes. The proposed system combines algorithms for local updates, cache management and server/client dialog. The local details management is an improvement of progressive reconstructions built on top of hierarchical structures. It enables fast, homogeneous accommodation and suppression of wavelet coefficients at any level of subdivision, with time complexity independent of the size of the reconstructed mesh. The cache structure wisely exploits the hierarchical character of the received data, in order to avoid redundant information transmission. The whole system enables the client to have total control on the quality of navigation according to its storage and processing capabilities, whatever the size of the mesh.

1. INTRODUCTION

Compression of 3D surface models is one of the main challenges in multimedia content transmission over networks. As scanning techniques allow the acquisition of larger and larger triangulated models with billions of polygons, it is still impossible to transmit these objects globally, even dramatically compressed. This is why such compression methods have to be progressive and adaptive when considered for online visualization. According to the restricted portions of the viewed model, only the pertinent data shall be transmitted, and with reduced quality if needed.

This requirement makes geometric wavelet techniques most appropriate for this type of application where transmission and visualization are performed one according to the other. Indeed, compression efficiency is of little interest if one has to transmit the whole model before reconstructing it, without exploiting the locality of the 3D walkthrough. The fact is that so called multiresolution (or multiscale) techniques not only are among the best in term of compression, but are also ideally suited for isolation of spatial as well as frequency components.

Together with very good animation possibilities, these are mainly

the reasons why this representation has been adopted in the international standard MPEG-4 Version 2, for 3D models encoding. This format, proposed in the context of AFX (Animation Framework eXtension) mainly relies on the adaptation of 2D wavelet image coding to second generation geometric wavelets[1][2]. The idea is to encode with minimum information non-significant portions of the object by exploiting variation similarities through levels of detail. The format is intended to allow complete adaptivity in transmission, either spatially or frequently.

However, while guaranteeing impressive compression rates and intrinsic adaptivity, this format specification does not imply any decoding efficiency or reconstruction speed, though it is a very important factor for a standard to be exploited. In this paper, we describe a complete system for real-time decoding and visualization of geometric data in a server-driven transmission scenario, for any wavelet scheme. It allows:

- real time decoding and reconstruction of the model with local, adaptive, periodic updates.
- exploitation of the particular wavelet bitstream to optimize the messages exchange and efficiently manage a cache on the client side.

This system is widely inspired by numerous earlier approaches in the domain of subdivision topology algorithms[3][4], multiresolution editing[5] and client/server-driven transmission systems.

The rest of the paper is organized as follows: in Section 1.1 we give an overview of previous work on related topic, and we list the main elements of our contributions in Section 1.2. In Section 2 we show how general algorithms can be derived from previous multiresolution geometry work. In Section 3 we describe the efficient insertion of these algorithms in a system including cache management, backchannel messages and multithreading. Results are given in Section 4.

1.1. Review of related work

Surface Compression is a necessary prior to any transmission, and even local storage when considering very large models. The particular mesh compression has been very widely studied in recent years, and one main distinction between methods has arose, according to whether the resulting representation is progressive or not. Non-progressive methods generally encode the topology and geometry separately but in one global pass. In progressive methods, the model is described by successive refinements of a coarse object, which makes the transmission process fully scalable. Thus, independently of the compression results, progressive methods have the advantage not to force full decompression of the model before visualization.

Multiresolution Representations A progressive representation of models can be *hierarchical*, i.e. built as a pyramidal scheme in which one can easily identify features at any level of detail and follow them throughout the progressive refinements. Such representations are perfectly suited for animation, adaptive reconstruction or multiresolution editing.

Subdivision Surfaces provide a particular representation of a class of models that can be regarded as the association of a coarse base mesh and a smoothing operator. This operator relies on canonical quadrisection of facets together with averaging of the resulting vertices. Numerous schemes have been proposed on which depend the properties of the limit surface, obtained after an infinite number of iterations of the smoothing operator. Although this representation can only represent a restricted class of surfaces, it provides a very convenient authoring tool as well as a very interesting representation for animation and multiresolution editing.

Geometric Wavelets can be seen as an extension to both subdivision surfaces[6] and classical multiresolution analysis theory. Classically, wavelets are functions expressing details with arbitrary small supports and large frequency. In this extension to manifold domains, the subdivision schemes are used as a starting point to a similar description; the quadrisections provide a notion of refinement for supports of wavelets, and geometric displacements naturally correspond to classical scaling spaces.

As this theory is actually a generalization to more general domains (typically polyhedra), the same post-transform processing used in classical theory can be transposed to geometric wavelets. For the particular case of wavelet coefficients compression, zero-tree algorithms can be applied as is to the geometric wavelet transform given a generalization of the notion of subbands[1].

1.2. Our contribution

By keeping in sight transmission and navigation quality, we gathered and improved algorithms for progressive reconstruction in order to allow optimal transmission and fully adaptive navigation. Thus, on top of data structures and algorithms related either to the progressive reconstruction or with multiresolution editing, we carried out a complete system of 3D content transmission, including fast reconstruction, efficient visualization, cache management and anticipation using backchannel messages. The main contributions of this paper leading to this transmission system are:

- the extension of real-time progressive midpoint wavelet schemes reconstruction to fully adaptive and local one. This gives rise to algorithms allowing incremental transform at any level of detail, performed locally and independently of the complexity of the displayed model.
- the extension of these algorithms to wavelets based on arbitrary subdivision schemes for use of the same functionalities.
- the demonstration of how simple client cache, server ordering of data and backchannel use can optimize the data transmission.
- the presentation of a complete multithreaded implementation, allowing independent processes for mesh reconstruction and 3D rendering and network dialogs.

In the context of the MPEG-4 Animation Framework eXtension, this system proves that in addition to its dramatic performance in term of compression, this encoding format is also suited

to real-time decoding for interactive visualization on low performance terminals.

2. ALGORITHMS OUTLINE

In a selective transmission scenario, wavelet coefficients arrive, are decoded, and then modify the currently reconstructed mesh by packets in the 3D scene which is updated according to the history of observer movements. Generally speaking, the coefficients describing a 3D model can be categorized whether they are necessary or not according to various criteria including visibility, geometrical details influence and terminal processing power.

Thus at a given moment, a reduced portion of the wavelet coefficients $(d_i^j)_{i,j}$ are used to build the model. It is then necessary to rebuild the model while taking into account on one hand the coefficients that became relevant and which it will be necessary to reflect, and on the other hand the coefficients that are not relevant any more and which it is necessary to remove from the representation.

The scene update is a process that is performed periodically and on which depends the whole navigation quality. For this reason, it is reasonable to require that the time spent to remove any coefficients should not be longer than that time spent to add them to the representation, and that the time of addition/suppression of coefficients must be independent of the size of the mesh being rebuilt.

In order to exhibit algorithms meeting these requirements, we start by solving this problem in the particular case of midpoint low pass filters, and extend the resulting algorithms to arbitrary subdivision schemes, interpolating or not. Starting from data structures and algorithms of [3], the first step is to allow fast and homogeneous suppression of wavelet coefficients. Let us consider the transition from a representation M_t of the surface by a set of coefficients E_t to a representation M_{t+1} defined by the set E_{t+1} . With the algorithms of [3], it does not raise any problem to adapt the coefficients that are in E_{t+1} but not in E_t . On the other hand, it remains to study how the coefficients which are in E_t but not in E_{t+1} can be suppressed efficiently from the current viewed mesh. Indeed it is impossible in real-time to rebuild the hierarchical representation from the start by using the whole E_{t+1} .

By naively considering removing a coefficient as adding its opposite, it appears that on one hand it is useless to proceed to the subdivision stage, and on the other hand a possible pruning stage may be necessary. Since the hierarchical influences of this suppression must be taken into account on all the levels of subdivisions, it appears difficult if not impossible to do otherwise. Moreover, the time saved by adding the opposite of a coefficient instead of carrying out an analysis is particularly perceptible in the case of approximating schemes, for which the analysis filters are generally not local.

The steps of the suppression of wavelet coefficients are as follows: first, symmetrically to adding coefficients, a loop is performed on every coefficient, changing the topology of the currently reconstructed mesh, and linking the coefficients to vertices of the newly subdivided mesh. Then a breadth-first traversal of the sub-meshes concerned by these coefficients in order to transmit perturbations from parent faces to children. The only non-symmetric stage is the pruning, in which facets are coarsened if their perturbations become negligible. As pruning is a priori a recursive process implying a depth-first traversal of the hierarchical structure, it is preferable to save time during the subdivision stage. For

this, we introduce a table containing, for i such as d_i is an active coefficient, node v_i indexing it. The localization of the nodes to be displaced is thus done in $O(1)$ time.

We can now consider a unified procedure to add as well as to withdraw wavelet coefficients on a partially rebuilt mesh. However, that is still not enough generalize from progressive to adaptive reconstruction. For this, it must be guaranteed that the modifications are performed locally on hierarchical structure, in the same way as it is done for coarsening.

The first stage in adding coefficients is the mesh subdivision in order to reflect all the coefficient supports. In the case of midpoint subdivision, mesh M_t can reflect these coefficients if and only if every scaling function ϕ_i^j corresponding to a coefficient d_i^j is linear on each facet of M_t . However, recall that the main criterion contributing to the navigation quality is the number of displayed facets. Consequently, we use instead an algorithm allowing to find the minimal mesh in term of a number of facets which enables the influence of the added coefficients. The algorithm consists in locating the scaling coefficients (displacements in this case) by their barycentric coordinates and recursively subdividing the mesh until it meets the linearity requirement. Regarding complexity, the completeness verification stage is replaced by a 3×3 matrix multiplication, i.e. 9 multiplications and 6 additions. The complexity class remains the same.

After having locally rebuilt the hierarchical representation, the actually displayed mesh still has to be modified. Rebuilding the facet list after each tree traversal is obviously not a good solution; the object actually displayed at a given time of visualization corresponds to the leaves of the arborescent facets representation. This representation must be extracted during each update from the displayed object. In the same way that it should be avoided to rebuild the whole hierarchical structure, this set of facets should be built only once and locally modified, in such a way that time spent to take into account (i.e. to remove or add) coefficients would only depend (ideally linearly) on the number of coefficients and not on the size of the mesh. Recall that vertices and facets are created and removed very often during navigation. Thus, if one does not want the newly created vertices to overflow available memory for representation, an additional process must be run, which manages these local modifications to keep the compact representation throughout the updates. Once again, in order not to increase time complexity, we introduce a moderate size structure which makes possible to check at the same time the requirements for locality and stability.

The algorithms exposed above are valid in the particular case of low-pass reconstruction filters that corresponds to a trivial subdivision scheme, in which the prediction is linear. In the general case, better prediction is obtained using more elaborate, classical subdivisions schemes, in particular Loop and Butterfly[7][8]. We next show how to generalize previous algorithms and study related issues.

Assume that we no longer use the linear wavelet basis of previous sections but a general scheme, interpolating or not. This scheme can be characterized by local masks describing low-pass matrix P^j and high-pass matrix Q^j . For example, in [1], Loop and Butterfly schemes are used for low-pass filtering.

The very simple algorithms of Section 2 which consist in assigning perturbations to each vertex and hierarchically propagating the corresponding displacements are not valid any more. Instead, it would be necessary to locally apply submatrices of P^j and Q^j corresponding to the refined zones, which is not very convenient since the coefficients can be added in arbitrary order and since

they have to be periodically propagated by packets to the descent. Fortunately the preceding algorithms can be generalized with the general settings in order to facilitate these two functionalities.

The idea is to consider the geometric offsets stored hierarchically and regarding hat functions no more like scaling coefficients but as absolute displacements of the control points. Thus, each time a vertex is created or moved at level j , it distributes its contributions with respect to the corresponding line of matrix $(P^j Q^j)$ to its neighbors. These contributions are stored according to their depth. It is thus guaranteed that the accommodation of coefficients can still be done locally in arbitrary order of subdivision.

The time complexity of the mesh update process is the maximum complexity of each one of its sub-task. All these algorithms will be detailed and proved in a forthcoming article.

3. NETWORK IMPLEMENTATION

Algorithms described in previous section are integrated in a classic server/client software architecture[9][10] implementing cache management, client/server dialog and multithreading suited to the particular form of the decompressed wavelet coefficients. The system is organized under server-driven scenario, according to the general idea that 1) the selection of the coefficients to be sent is made by the server, which reduces terminals task, whose capacities are not guaranteed 2) the customer sends information on its cache state to minimize the multiple transfers of the same data.

The system architecture is organized as follows: the data present on the server are stored in a form containing zero-tree representation suggested in MPEG/AFX[11], encapsulated with markers allowing fast localization of regions of interest on the transmitted object, as well as the level of precision (number of bitplanes). Recall that with the zero-tree representation coefficients are sent in their increasing magnitude order, starting with the most significant bits, interleaving information needed to connect this sorting to the underlying hierarchical structure. On the client side, after decoding of the wavelet coefficients, those are stored according to the edges (indexing trees) of the base mesh they refine, and for each edge, they are sorted by order of magnitude. Indeed the resulting screen-based error is proportional to the coefficients magnitude.

As exposed above, the server is assumed to be responsible for data selection according to upstreaming information. Thus, it has to be able to locate efficiently the portions of its binary representation corresponding to either regions of interest or different refinement type (additional wavelet coefficients, further bitplane refinement).

Client cache management classically consists in dealing with available memory space to keep the most relevant data in order to minimize multiple transmission of the same data: if a portion of the scene has already been visited, it is useless to request the corresponding data to the server if they are still present in memory. As 3D scenes are generally very large, the client cannot keep all received data, and has to periodically discard refinement information. When the cache is full, the selection of the elements to be removed must be carried out by considering statistics on the visited areas. For example, if the client often returns to visit a given part of the object, it may naturally stay longer in memory than an area visited only once.

As stated above, the considered scenario is server-driven, meaning that the choice of relevant information to transmit is performed by the server, according to the client navigation parameters. These parameters are sent regularly through a back-channel as viewpoint

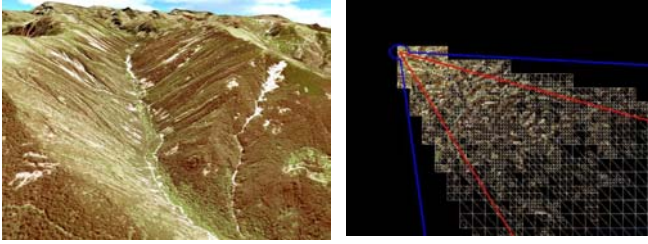


Fig. 1. View-dependent refinement of terrain database. Left: perspective view. Right: reconstructed refinements according to cache content.

and view direction (and optionally view angle), letting the server know which facets of the base mesh are seen, and the distance at which they are seen. It can then decide the trees to be sent and at which precision, in term of number of bitplanes.

The use of the multi-threading, for applications of interactive visualization, allows to smooth navigation by distinguishing the various tasks carried out and by performing them asynchronously. Each of them is performed autonomously and does only require casual non-blocking interactions with the other processes.

4. RESULTS

Since in multi-threading configuration the mesh updates are done by a separate thread, the actual result criterion may not be the frame rate at which navigation occurs, but instead the time taken to refine or coarsen the mesh according to packets of newly relevant/non-relevant wavelet coefficients. Taking the *Feline* (Figure 2) as test model, 64000 midpoint wavelet coefficients are gathered in 690 ms, taking the mesh from 504 facets to 128589 facets on a 1.4 GHz processor. Note that such a large update is unlikely to be performed in one pass in real navigation. In the same conditions, 32000 coefficients are gathered in 160 ms, taking the model from 504 to 32256 facets. The converse process for the same coefficient packets are performed in 631 and 150 ms respectively, showing that consistency between addition and removal of coefficients is guaranteed by algorithms explained in of Section 2.

The network implementation including server data sorting, client cache management and backchannel messages enables fully view-dependent transmission thus reducing considerably the amount of transmitted data for close navigation models. As an example, using these algorithms on large terrain data allows low altitude navigation on arbitrary large terrains with a 56K modem. Figure 1 shows navigation over France.

5. CONCLUSION AND FUTURE WORK

In this paper, we gave the outlines of algorithms any decoder/player could use to exploit very efficiently the compressed format for surfaces encoded by geometric wavelets. We showed how this representation can be practically used to fit to the terminal capabilities, and how client-server dialog and cache management can further reduce the amount of transmitted data for the same rendering quality.

In future work, we plan to use this framework as a base for complete evaluation of QoS for compound AFX applications including view-dependent tools as VTC (Visual Texture Coding) and

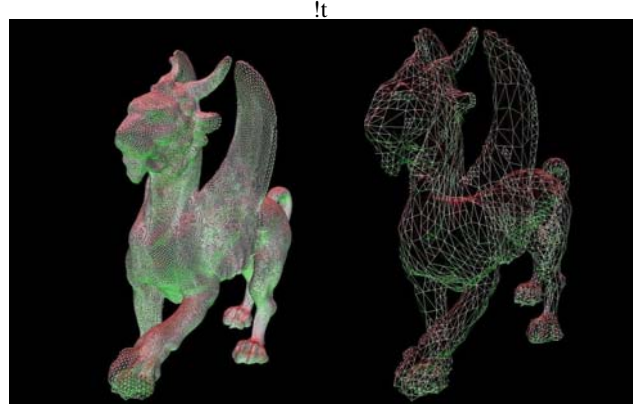


Fig. 2. Adaptive reconstruction of the Feline model

the AFX generic backchannel, in terms of adaptivity to both network properties and terminal processing power.

6. REFERENCES

- [1] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *SIGGRAPH 2000 Conference Proceedings*, 2000.
- [2] Francisco Moràn and Narciso Garcia, "Hierarchical coding of 3d models with subdivision surfaces," September 2000, pp. 113–120.
- [3] A. Certain, J. Popovic and T. DeRose and T. Duchamp, D. Salesin, and W. Stuetzle, "Interactive multiresolution surface viewing," *SIGGRAPH 96 Conference Proceedings*, pp. 91–98, 1996.
- [4] K. Pulli and M. Lounsbery, "Hierarchical editing and rendering of subdivision surfaces," Tech. Rep. TR-97-04-07, 1997.
- [5] D. Zorin, P. Schröder, and W. Sweldens, "Interactive multiresolution editing," July 1997.
- [6] Michael Lounsbery, Tony D. DeRose, and Joe Warren, "Multiresolution analysis for surfaces of arbitrary topological type," *ACM Transactions on Graphics*, vol. 16, no. 1, pp. 34–73, Jan. 1997, ISSN 0730-0301.
- [7] C. Loop, *Smooth Subdivision Surfaces Based on Triangles*, Master's thesis, University of Utah, Department of Mathematics, 1987.
- [8] Dyn N., Levin D., and Gregory J. A., "A butterfly subdivision scheme for surface interpolation with tension control," *ACM Transactions on Graphics*, vol. 9, no. 2, pp. 160–169, April 1990.
- [9] M.R Macedonia, M.J Zyda, D.R Pratt, P.T Barham, and S Zeswitz, "Npsnet : a network software architecture for large scale virtual environment," *Presence*, vol. 3, no. 4, pp. 265–287, 1994.
- [10] S. Singhal and M. Zyda, *Networked virtual environments*, ACM press, 1999.
- [11] Mikael Bourges-Svenier, Iver Grini, Michael Stelias, and Dominique Curet, "Pdram of iso/iec 14496-1 / amd4 n4415," *ISO/IEC/JTC1/SC29/WG11 MPEG2001*, Pattaya, December 2001.