WAVELET COMPRESSION OF SEMI-REGULAR TETRAHEDRAL VOLUME MESHES

Silvia M. Castro[1], Liliana R. Castro[2]

[1]Universidad Nacional del Sur, Dpto. de Ciencias e Ingeniería de la Computación, Avda. Alem 1253, B8000CPB Bahía Blanca, Argentina
[2]Universidad Nacional del Sur, Dpto. de Matemática
Avda. Alem 1253, B8000CPB Bahía Blanca, Argentina

smc@cs.uns.edu.ar (Silvia M. Castro)

Abstract

In this article we introduce a new compression scheme that can be effectively used to compress volume data sets by exploiting a multiresolution model based on a semi-regular tetrahedral mesh, that is a mesh that is regular except on the coarsest level. In order to generate the multiresolution representation, we use a wavelet based approach that allows compression and progressive transmission. Starting with an initial semi-regular tetrahedral mesh Γ_∞ and successively applying the wavelet transform, we construct coarser representation levels of the given mesh. At the end, a coarse base mesh Γ_0 together with a sequence of detail coefficients are obtained from the decomposition of the mesh at these different levels of detail. In order to do that, we use a Haar-like wavelet basis defined over a tetrahedron as the first step for defining this kind of bases over an object represented by tetrahedra. The considered base mesh is represented at the lowest resolution and it does not have the connectivity subdivision property. The obtained result is a hierarchical data description suited for compression. In addition, we can obtain higher levels of compression using the lossy and lossless compression algorithms proposed in this paper. In the case of transmission, we have analyzed a protocol that allows progressive transmission of the mesh.

Keywords: 3D Compression, Wavelets, Semi-Regular Meshes.

Presenting Author's biography

Liliana Castro received a bachelor and a Master in Mathematics in 1979 an 1982, respectively and the Doctorate in Control Systems in 2001 at the Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina. Since 1999 she is Associate Professor at the Departamento de Matemática, UNS and she is also with the Instituto de Investigaciones en Ingeniería Eléctrica *Alfredo Desages*. Since 2000 she is head of the Modeling Group at the Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica, Departamento de Ciencias e Ingeniería de la Computación, UNS. She teaches undergraduate introductory courses on multivariate analysis, a graduate course on wavelet analysis and a graduate course on functional analysis. Her research interests include nonlinear system modeling, identification, piecewise linear approximation of nonlinear systems and geometric modeling using wavelets.

1 Introduction

Tetrahedral meshes are well suited for irregular sampling and multiresolution analysis; they can be used to model an object given by sparse data, and this mesh is a general topological domain for the intrinsic representation of the volume. Due to the good well known properties of tetrahedral meshes, they are the natural choice for volume data representation.

In most application areas using tetrahedral meshes, data can be attached to different mesh elements. In this way, data can be attached to vertices, edges, faces, border faces or tetrahedra. For example, the material identifiers might be attached to the tetrahedra, the density to the vertices or to the tetrahedra, the intensity of a flow to the edges. The tetrahedral mesh also serves to parameterize the domain of a function that can be a scalar one.

In this paper we propose a volumetric model based on wavelets that allow compression and progressive transmission. We also present the whole compression pipeline, including the complete description of each stage. For lossy compression, how to control the compression error is described. Our compression algorithm can be extended in a natural way to support compression of different kind of data functions defined on different types of mesh elements.

2 Related work

Chui [1] reported that the comparison among several 2D lossy compression techniques shows that methods based on wavelet transform (WT in short) are the best ones. The idea of using a three dimensional wavelet to approximate three dimensional volume datasets was introduced by Muraki ([2], [3]). He constructs a 3D orthonormal wavelet basis using all possible tensor products of one-dimensional basis functions and presents the potential of the 3D WT for volume visualization but he did not mention whether the encoding technique actually reduces storage space.

In [4] is presented a wavelet-based 3D compression scheme for very large volume data. This is an effective 3D compression scheme that exploits the power of wavelet theory; the definition of the wavelets is also based on the tensor products of one dimensional wavelets.

Although this methodology gives a simple way for wavelet constructions, it cannot be used without introducing degeneracies when representing surfaces or volumes defined on general domains of arbitrary topological type, like spherical domains. In order to compress tetrahedral meshes using wavelets, it is first necessary to define the wavelets on arbitrary topological domains on R^3.

Lounsbery [5] and Stollnitz *et al*. [6] were the first who introduced wavelets from a different point of view, defining them on arbitrary topological domains on R^2. This approach was generalized *a posteriori* by Sweldens ([7], [8]) who recognized that the *lifting scheme* he proposed was a generalization of Lounsbery's methodology. Other wavelet constructions based on subdivision to represents functions defined on spherical triangles defined for spherical domains were introduced by Schröder and Sweldens [9], Nielson *et al*. [10], Bertram *et al.* [11], and Bonneau [12].

In [13] and [14], other techniques for representing wavelet based volume data are given. It is clear that in order to have a compression algorithm based on wavelets, it is necessary to define the underlying wavelets. Then if we want to compress tetrahedral meshes based on wavelets we first need to have wavelets defined over tetrahedra.

In this article, we introduce a new compression scheme based on wavelets that can be used to compress semi-regular tetrahedral meshes. In order to do that, we use the Haar-like wavelet basis defined over a tetrahedron [4] as the first step for defining this kind of bases over an object represented by tetrahedra. The paper is organized as follows. In Section 3, we introduce the wavelets basis defined over a tetrahedron, the model and its respective Data Representation. In Section 4, we provide a detailed description of our compression scheme. Finally, in Section 5 we present the conclusions and directions for future work.

3 Wavelets volumetric model

The mesh representing the object must store the 3D geometry, its topology and its attributes. One of the main advantages of tetrahedral meshes is that any other polyhedral mesh can be reduced to a tetrahedral one; hence a tetrahedral mesh can represent a volume with arbitrary topological type. Then, beginning with a tetrahedral mesh and using the subdivision and the defined wavelets, we will show how to generate a model of a volumetric object of arbitrary topological type that can be used for compression and progressive transmission.

3.1 Wavelet basis defined over a tetrahedron

To define the wavelets we adopt the subdivision method defined in [15], based on the recursive subdivision of the tetrahedron. We have chosen this method because it has the property that the subdivision of any given tetrahedron results in a group of elements of at most three congruence classes, no matter how many successive refinement steps are performed.

Beginning with a tetrahedral net and using this subdivision method, it is possible to construct a wavelet basis following the process given by Girardi and Sweldens [16]. Also, it is possible to calculate the coefficients related to the analysis and synthesis processes using the fast WT defined in [4].

3.2 Volumetric model

The proposed model is based on a semi-regular tetrahedral mesh, *i.e.* a mesh that is regular except on the coarsest level. This kind of mesh is especially well suited for different multiresolution algorithms. A semi-regular tetrahedral representation is a sequence of approximations at different resolution levels. The corresponding sequence of nested refinements is transformed using the WT to a representation that consists of a coarse resolution or base mesh and a set of detail coefficients that represents the differences between successive resolution levels. The base mesh Γ_0 is the mesh at the coarsest resolution and does not have the subdivision-connectivity property. Then, the model consists of a base mesh and a sequence of modifications. These modifications correspond to terms that locally capture the details of the object at different resolutions.

Hence, the developed model begins with the finest resolution mesh Γ_∞ and decomposes it on the coarsest mesh Γ_0, together with a set of detail tetrahedra generated during the analysis. So the multiresolution representation of the volume consists of the base mesh and the whole set of details (

Fig. 1).

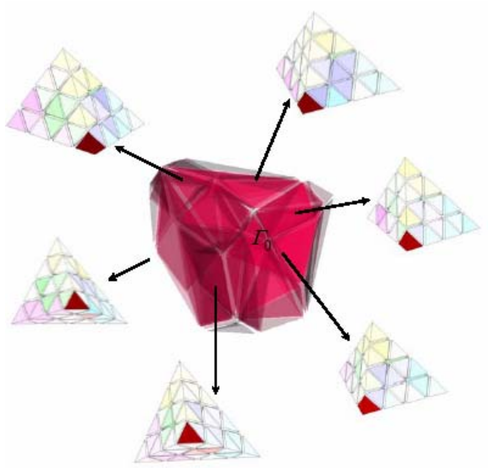

Fig. 1. Multiresolution Representation of the Volume

Taking into account that the WT concentrates the energy of the object on the coarsest resolution mesh and that the mesh has space localization, this model is suited for compression. However, the compression will depend not only on the chosen wavelets but also on the following issues:

- The number of coefficients needed to achieve a good approximation tothe volume.
- The mesh encoding and storage with the minima number of bits.

If we transmit the base mesh and all the details, the compression method can be considered as a lossless compression encoding. If not all the coefficients are stored, it can be considered a lossy compression encoding. As the highest energy concentration is achieved in the lowest resolution mesh, only between the 10% and 15% of the detail coefficients are required in order to have a good approximation to the real volume.

To transmit the underlying mesh of this model *via* Internet, we must consider the transmission of the base mesh and the detail coefficients. A robust transmission has to guarantee that the base mesh is completely transmitted before the details are transmitted and added. After transmitting the base mesh, the details must be sent and added to it.

The transmitted details can be added to the base mesh all at once after they have been received or one by one as long as they are received, until certain requirements are fulfilled. This last option allows the progressive transmission of the model.

3.3 Encoding wavelet coefficients: the data structure

The data structure proposed for our model consists of the data structure for the base mesh $\Gamma_0=<\sigma_0, \sigma_1, ..., \sigma_n>$ and of a forest of octrees to store the details corresponding to each one of its cells. (

Fig. 2).

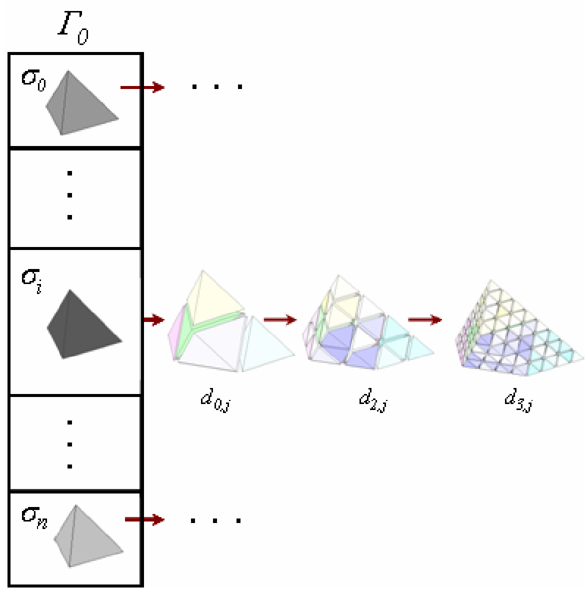

Fig. 2. Data Structure

Fig. 3 shows a tetrahedron of the base mesh and its associated details represented by a forest whose trees have the coefficients $d_{0,j}$ as roots. Each tree in the forest is a hierarchy of regular tetrahedra and the relation of dependency is structured as an octree of tetrahedra. As every cell vertex define a regular grid, the coordinates of each vertex can be retrieved from its position on the grid.

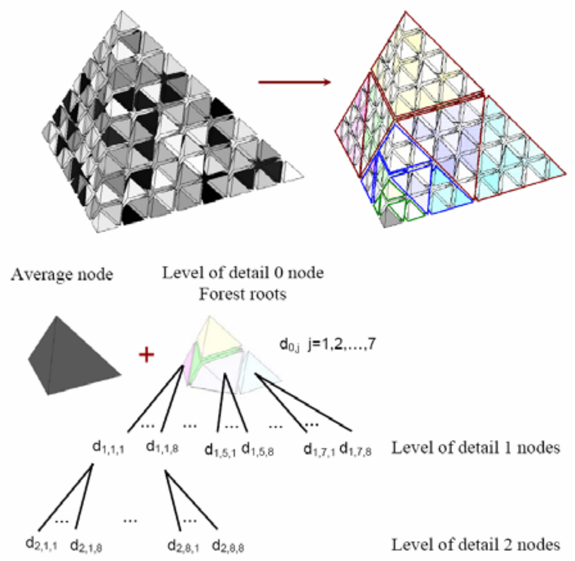

Fig. 3. Wavelet decomposition of a tetrahedron and graphical representation of that decomposition (base mesh+forest)

This multiresolution data structure is generated from a given volume with the connectivity-subdivision property Γ_J, being J the maximum resolution level. The WT is performed on each set of tetrahedra that replaces the tetrahedron Γ_J until the lowest resolution tetrahedra Γ_0 is obtained. The set σ_0 coincide with a cell of Γ_0 and its forest of details (

Fig. 4).

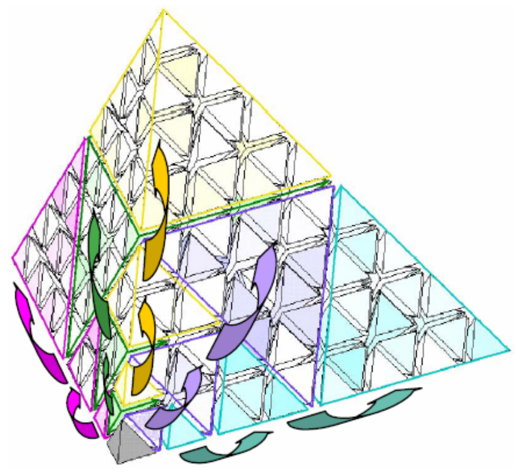

Fig. 4. Base Mesh Tetrahedron and the Forest of Details

A forest, the decomposition level and a key to a reference coordinate corresponding to the lowest resolution tetrahedron are stored in a heap for each cell of the base mesh. The reference coordinate will be used to retrieve the geometry of the tetrahedron corresponding to those ones obtained from it at a finer resolution.

3.4 Space complexity of the data structure

The mesh is stored using an indexed structure like winged edge that encodes, for each tetrahedron, the indices of its vertices and the adjacent tetrahedra, along with the four faces. The total storage cost corresponding to the data structure of the reference mesh can be calculated in the following way:

TotalCost = ConnectivityCost + GeometryCost + AttributesCost.

If n is the number of vertices of the reference mesh and t is the number of tetrahedra, the amount of t tetrahedra is around $6n$. The connectivity cost requires store $4t$ indices (one for each vertex), $4t$ indices corresponding to the adjacent tetrahedra and $3n$ vertex coordinates. Since the cost of a scalar attribute is t, if we consider that we have only one scalar attribute, the storage cost is:

TotalCost = 8t + 3n + t = 48n + 3n + 6n.

Assuming 4 bytes for the indices, 2 bytes for each coordinate and 2 bytes for a scalar attribute, the storage cost in bytes is *210n bytes.* From this, it is clear that the connectivity information dominates the storage cost and it must be compressed.

The storage cost of our model is:

TotalCost = BaseMesh Cost + ForestCost.

The storage cost of the base mesh is the storage cost of a winged edge data structure. So,

BaseMesh Cost = 210n_{bm} bytes,

being n_{bm} ($n_{bm} << n$) the amount of vertices of the base mesh. Then:

TotalCost= 210n_{bm} bytes + ForestCost.

Each tetrahedron of the base mesh has a forest associated to it and any other node describes a detail tetrahedron. The eight corresponding sons are the tetrahedra obtained from Bey's subdivision method [17][17].

Each one of the seven trees of the forest is a complete octree that we can implicitly codify; *i.e.* we do not need to store the connectivity and the structural information; since we have regular tetrahedra, the vertex coordinates are implicit. As a consequence, in order to encode the details we only need to encode the field or the attribute values.

We have supposed that each attribute is stored using 2 bytes. These attributes are stored for each tetrahedron of the base mesh and these values have been taken into account in the required storage space for the base mesh. The number of detail tetrahedra plus the tetrahedra of the base mesh is the number of tetrahedra of the reference mesh. However, for each of them we only store the detail corresponding to the attribute. Then we have $t - t_{bm}$ detail tetrahedra and the storage cost becomes:

TotalCost = $204n_{bm}$ bytes + $t - t_{bm}$

= $210n_{bm}$ bytes + $(6n - 6n_{bm})$byte.

The storage cost is significantly reduced compared to the total cost of the reference mesh ($n_{bm} << n$). Instead of the winged edge data structure, for a manifold tetrahedral base mesh can be used the Half–Face (CHF) Data Structure [18]. If large datasets would be represented with the base mesh this could not be appropiate and we should use a LOD Data Structure [19].

3.5 How the model allows compression

This model allows lossless and lossy compression. We will apply the compression to a scalar function defined on the tetrahedra and we will show how the cell based scheme allows to achieving a high compression level. We can have two different alternatives to compress the volume: one is to reduce the number of coefficients to approximate the volumetric data and the other is to encode and to store the necessary information using a small number of bits. We have implemented the first option and have left the second one for future work.

3.5.1 Two models for compression

After applying the WT on the given data, the multiresolution model is obtained. Then, the attribute values associated to the tetrahedra are uncorrelated and the energy of the original data is concentrated in a relative small number of coefficients.

The key behind wavelet lossy compression scheme is to select the coefficients with smallest norm and replace them by zero. This criterion minimizes the L^2 norm of the resulting approximation error. No matter which criterion is selected to set the detail coefficients to zero, the original signal will be approximated with a very small number of nonzero coefficients. Then, we can obtain compression in two different ways:

All coefficients that remain in the representation are encoded with a lower amount of bits per coefficient using run-length encoding, vectorial quantization or differential encoding.

A very small portion of coefficients (between 10% and 15%, for example) are kept without modification and the rest of them are set to zero. Hence, it would be reasonable to keep only the non-zero coefficients. In order to do this, we can take advantage of the spatial localization property of the wavelets: the behavior of the detail coefficients of a father in a given forest tree allows predict the behavior of its descendents.

3.5.2 Error control

In the previously described models we have supposed that we could control the level of compression by specifying a given percentage of coefficients that are not set to zero (surviving coefficients). In both cases, we can control the number of surviving coefficients by specifying an adequate threshold and set to zero all coefficients whose magnitudes are smaller than it. This threshold can be automatically determined taking into account the maximum allowed approximation error. The ideal way to compute the threshold is by sorting all the coefficients in decreasing order of significance. However, when the amount of data is huge, this is impractical ($O(n \log n)$). Then, if we want to control the error, we should find the threshold without sorting the data.

One can also specify a threshold and encode all the coefficients of magnitude greater than it and eliminate all the other ones ($O(n)$). In this case, even if the approximation error can be computed, it can not be controlled since a fixed number of coefficients are eliminated, depending on their value respect to the threshold.

Finally, the compression scheme developed so far allows compression of non structured volumes decomposed in atomic tetrahedral elements and that have scalar or vectorial values defined on them. It is then necessary to consider the appropriate metric depending on the nature of data, *e.g.* geometric, color, texture data, *etc*. In general, the L^2-norm is considered.

3.5.3 Decompression

The decompression allows to reconstructing the received information of the progressive transmission. The base mesh will be received first and will be decoded according to the encoding method. Once the reconstruction of the base mesh is completed, the inverse WT will be applied to the detail coefficients received afterwards.

3.6 How the model allows progressive transmission

For the mesh transmission we the so-called *mesh transfer protocol (MTP)* defined by Staadt [20], which is a modification of the protocol for the transmission of semi-regular meshes. In order to guarantee a reliable and ordered delivery of the base mesh to destination, we use TCP for the transmission. To do this, the protocol must use a lot of overhead communication; fortunately the base mesh is small compared to the finest resolution level.

After the transmission of the base mesh has been completed, the details must be sent. Since the position information of the details is implicitly given by the order in the chain, they are sent using the TCP protocol.

4 Compression pipeline

The pipeline describes all the necessary steps to generate a compressed data representation beginning with a high resolution volume, the transmission of the data and the decompression of the compressed data to finally obtain the reconstructed volume. In this pipeline, described in Fig. 5, the compression and the decompression stages can be clearly identified.

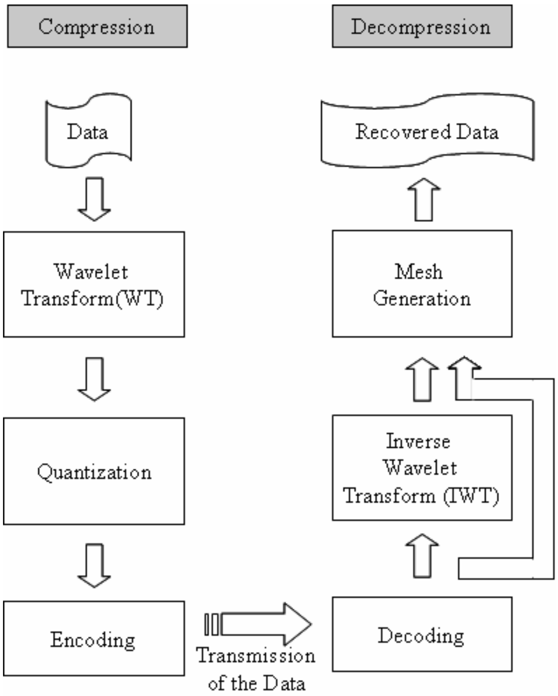

Fig. 5. Compression Pipeline

The proposed wavelet-based coding algorithms produce embedded data streams that can be decoded up to a lossless level.

The decompression pipeline inverts the process to obtain the reconstructed data.

4.1 Compression

A typical wavelet compression algorithm has three basic components: transformation, quantization and encoding.

Because of the order of magnitude of the attribute data, the transformation process might vary from dataset to dataset and from application to application. Previous to the application of the WT, we normalize the data between 0 and 1. The transform generates the described multiresolution data representation, *i.e.* the octree forest. The attribute values associated to the tetrahedra are then uncorrelated and the energy of the original data is concentrated on a relative small number of coefficients.

Before sending the coefficients, we send the encoded base mesh. Afterwards, all the wavelet coefficients are sent.

At the quantization step, different procedures can be applied depending on the lossy or lossless used scheme.

In the lossless compression case, the coefficients are not quantized and must be coded only before transmission; the data will be coded with the Δ-encoding.

In the lossy case, the wavelet coefficients are quantized. After this, and before encoding them, the coefficients less than a thresholdμ, must be set to zero and encoded afterwards. As we can assume that many coefficients will be equal to zero, we can take into account the proposal presented in [20]. Then all non-zero coefficients are represented by two-tuples, where the first element represents the number of bits required to encode the second one while the second element contains the data value itself. All negative numbers are replaced by their absolute values and the first bit indicates the sign. The zero coefficients are encoded with the run-length encoding because we relative long runs of zeros is expected.

In order to transmit the data, they should be merged from the octree into a bitstream; *i.e.* the coefficients in the octree forest must be mapped into a 1D array. This step must traverse the data representation taking into account that the most significant coefficients should be sent first. Then the order to convert the data from the octree into the bitstream must be established.

Considering the usual spatial coherence in the data, it is quite possible that the zero or insignificant coefficients exist in clusters. Then we must take this into account to establish the order to merge all the coefficients in streams.

Fig. 2 illustrates how the base mesh and the wavelet coefficients are stored. In this figure we have considered a decomposed unit block (an octree), at level-three of the multiresolution representation of the corresponding original unit block.

To generate the bitstream, we traverse the forest by level and from left to right in each level. In Fig. 6 we see how we generate the bitstream from the data representation.

4.1.1 Error control

In the lossy compression case, once all the insignificant values are set to zero, how to control the error must be considered. Given a threshold $\mu > 0$, the detail coefficients d_{ij} such that $|d_{ij}| < \mu$ are set to zero, the L^2 -error is given by:

$$\varepsilon = \sum_{|d_{ij}|<\mu} d_{ij}^2 .$$

As well as in the lossless case, the higher coefficients are the most important ones. In addition, the insignificant coefficients must be replaced by zero in order to have an approximation with a smaller amount of bits. We can do this in two different ways.

We can specify a threshold $\mu > 0$ and set to zero all the coefficients smaller than this threshold. In this case, the error will be determined by all the coefficients that are put to zero with respect to the total number of non-zero coefficients.

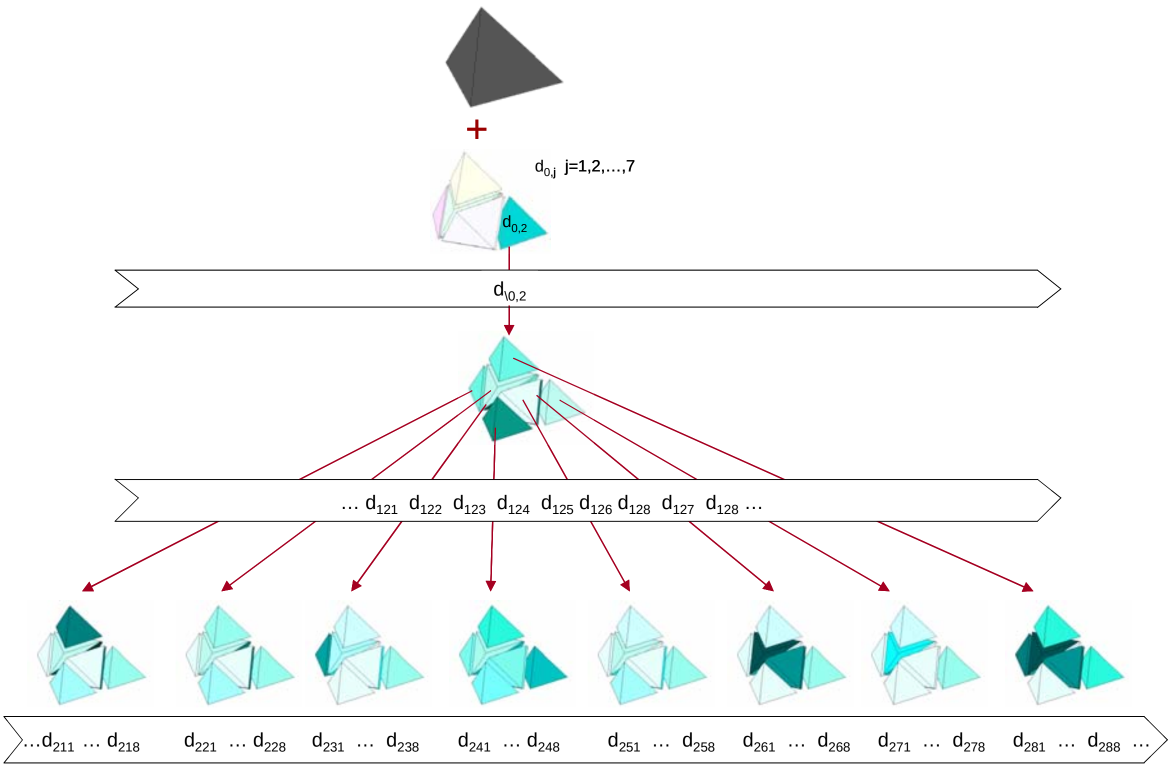

Fig. 6. Transmission of the Details

Another way to do this, consist on specifying a percentage of coefficients that must be set to zero; in this case, the threshold must be calculated and in order to obtain a good reconstruction the biggest wavelets coefficients must be kept. That is, if we specify the percentage *p* of coefficients that must survive, we can calculate the threshold sorting the absolute value of the coefficients in decreasing order, then find the coefficient at its position *p x #total of coefficients* and discard the rest. Since this method is not practical for large volumetric datasets, we only consider the ther way.

We use each octree group as a block and use this distribution in order to calculate a good threshold. We calculate, for each block *i* that contain at least one non-zero coefficient, the quotient

$$\eta_i = \frac{\#\, of\ coefficients \neq 0}{\#\, of\ all\ coefficients},$$

that gives a measure of how quickly the tetrahedron value changes in the block. It is reasonable that more non-zero coefficients are taken from blocks with higher η_i.

If the maximum level of decomposition is *J*, each block has 8^J tetrahedra and each block has 8^J coefficients. Then, if the volume has $t_{rb} \times 8^J \times \hat{\eta}$ coefficients, these must be distributed in $t_{rb} \times 8^J \times \#\, of\ coeff \neq 0 \times \eta_i \big/ \sum_j \eta_j$ per block. For the block *i*, we assign to the block threshold the value of

the coefficient $\mu_i = \eta_i \big/ \sum_j \eta_j \times t_{rb} \times 8^J \times \#coeff \neq 0$, where the weight $\eta_i \big/ \sum_j \eta_j$ is a relative measure of the data complexity. Then, the μ_i*-th* biggest coefficient is the block threshold value and all the coefficients lower than this threshold will be set to zero. Then, if we choose to keep a given percentage of coefficients, we will not take the same number of coefficients for each block. This allows to keeping more coefficients on the regions with greatest changes.

4.2 Decompression

The reconstruction can be described in four stages. In the first stage, the base mesh must be completely decompressed. At this time, the base mesh can be reconstructed and rendered as well.

After that, the wavelet coefficients will arrive and can be decoded as long as they come. As soon as the details arrive, they are stored using the data representation. This allows the mesh to be rendered progressively; this can be updated at given intervals of time.

5 Conclusions and future work

In this paper we show how to use a wavelet based model in order to have compression and progressive transmission. We present the complete compression scheme for tetrahedral meshes. For the compression stage we have also shown how to reduce the number of coefficients needed for approximating the volumetric data and how to encode the information according to the proposed model. In the case of transmission, we analyzed a protocol that allows progressive transmission of the mesh.

Some topics deserve further investigation. We have decided to encode all the wavelets coefficients; another possibility consists on removing the insignificant coefficients and the approximation of the original data will be reconstructed only with the significant ones. In this case, we also need the positional information. Some approaches exists but the compression scheme is for Regular Volume Data

Our framework is also an important initial work to construct multiresolution representations of irregular meshes. Future work includes the extension of our results to functions defined on unstructured tetrahedral domains and also the representation of the geometry of the underlying domain. These extensions would allow obtaining a wavelet-based method to model irregular tetrahedral meshes without the subdivision connectivity property.

6 References

[1] C. Chui. *An Introduction to Wavelets.* Academic Press Inc., 1992.

[2] S. Muraki. Approximation and rendering of volume data using wavelet transform. In *Proceedings of Visualization '92,* 21-28, 1992.

[3] S. Muraki. Volume data and wavelet transform. *IEEE Computer Graphics and Applications*, 13:50-56, 1993.

[4] S. Castro, L. Castro, L. Boscardín, and A. De Giusti. Multiresolution Wavelet Based Model for Large Irregular Volume Data Sets. *WSCG2006 Full Papers Proceedings*, 101-108, 2006.

[5] J. Lounsbery. Multiresolution Analysis for Surfaces of Arbitrary Topological Type. PhD dissertation, University of Washington, Washington, Seattle, 1994

[6] E. Stollnitz *et al*. Wavelets for Computer Graphics: Theory and Applications. Morgan & Kaufmann Publishers, Inc, 1996.

[7] W. Sweldens. The Lifting Scheme: A new Philosophy in Biorthogonal Wavelet Constructions. Proceedings of the SPIE 2569, pp.68-79, 1995.

[8] W. Sweldens. Sweldens, W. The Lifting Scheme: A Custom-design of Biorthogonal Wavelets. Applied and Computational Harmonic Analysis 3, pp.186-200, 1996.

[9] P. Schröder and W. Sweldens. Spherical Wavelets: Efficiently representing Functions on the Sphere. ACM Proc. SIGGRAPH, pp. 161-172, 1995.

[10] G. Nielson *et al*. Haar Wavelets over Triangular Domains with Applications to Multiresolution Models for Flow over a Sphere. Proc. of IEEE Visualization, pp. 143-150, 1997

[11] M. Bertram *et al*. Bicubic Subdivision-surface Wavelets for large Scale Isosurface Representation and Visualization. ACM Proc. SIGGRAPH, pp. 389-396, 2000.

[12] G. P. Bonneau. Multiresolution Analysis on Irregular Surface Meshes. IEEE Trans. Visualization and Computer Graphics, vol.4, pp. 365-378, 1998.

[13] U. Labsik. Multiresolution Mesh Processing for Triangular and Tetrahedral Meshes. PhD dissertation, Der Technischen Fakultät der Universität Erlangen-Nürnberg, 2003.

[14] L. Linsen *et al*. Hierarchical Large-Scale Volume Representation with $\sqrt[3]{2}$ Subdivision and Trivariate B-Spline Wavelets. TR CSE-2002-7, Department of Computer Science, University of California, Davis, 2002

[15] P. Magillo, and L. De Floriani. Multiresolution Mesh Representation: Models and Data Structures. In Tutorial on Multiresolution in Geometric Modelling, Spriner Verlag, pp.363-416, 2002.

[16] M. Girardi, and W. Sweldens. A new class of unbalanced Haar wavelets that form an unconditional basis for Lp on general measure spaces. Technical Report 1995:2, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995. http:// citeseer.ist.psu.edu/girardi95new.html

[17] J. Bey. Tetrahedral Grid Refinement. *Computing*, 355-378, 1995.

[18] M. Lage, T. Lewiner, H. Lopes and L. Velho. Topological Data Structure for Tetrahedral Meshes. Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05), 2005.

[19] E. Danovaro and L. De Floriani. Half-Edge Multi-tessellation: A compact representation for

multiresolution tetrahedral meshes. In Proc. 1st International Symposium on 3D Data Processing Visualization and Transmission, 494-499, IEEE Computer Society, 2002.

[20]O. Staadt. Multiresolution Representation and Compression of Surfaces and Volumes. PhD dissertation, Swiss Federal Institute of Technology Zürich, 2001.

[21]I. Ihm and S. Park. Wavelet-Based 3D Compression for Very Large Volume Data. *Graphics Interface*, 107-116, 1998.

[22]D. Blanford, G. Blelloch, D. Cardoze and C. Kadow. Compact Representations of Simplicial Meshes in two and three Dimensions. Proc. 12th International Meshing Roundtable (IMR), 135-146, 2003.