Wavelet Based Multiresolution Analysis of Irregular Surface Meshes

Sébastien Valette, Rémy Prost

Wavelet-Based Multiresolution Analysis of Irregular Surface Meshes

Sébastien Valette and Rémy Prost, *Member, IEEE*
CREATIS*, Lyon, France

***Abstract*— This paper extends Lounsbery's multiresolution analysis wavelet-based theory for triangular 3D meshes, which can only be applied to regularly subdivided meshes and thus involves a remeshing of the existing 3D data. Based on a new irregular subdivision scheme, the proposed algorithm can be applied directly to irregular meshes, which can be very interesting when one wants to keep the connectivity and geometry of the processed mesh completely unchanged. This is very convenient in CAD (Computer Assisted Design), when the mesh has attributes such as texture and color information, or when the 3D mesh is used for simulations, and where a different connectivity could lead to simulation errors. The algorithm faces an inverse problem, for which a solution is proposed. For each level of resolution the simplification is processed in order to keep the mesh as regular as possible. In addition, a geometric criterion is used to keep the geometry of the approximations as close as possible to the original mesh. Several examples on various reference meshes are shown, to prove the efficiency of our proposal.**

***Index Terms*— irregular meshes, wavelets, multiresolution.**

I. Introduction

MULTIRESOLUTION analysis of 3D objects is receiving a lot of attention nowadays, due to the practical interest of 3D modeling in a wider and wider range of applications, such as Computer Graphics and Computer Assisted Design (CAD). Multiresolution analysis of these objects gives some useful features : several levels of detail can be built for these objects, accelerating the rendering when there is no need for sharp detail, and allowing progressive transmission. Another feature is that multiresolution analysis can be an efficient means of data compression. A survey of the existing methods used to simplify meshes, which is the first step for processing multiresolution analysis, was reported in [7]. We can briefly mention vertex decimation [15], edge contraction [8], wavelet based analysis [14], valence-based mesh simplification [1], [4]. We concentrated on the wavelet-based method, because wavelets are well-suited for multiresolution analysis. In [2] an algorithm able to perform multiresolution analysis on irregular meshes is proposed, but it is restricted to planar or spherical meshes. The algorithm described in this paper can be applied to any manifold mesh, with any genus and any number of boundaries. In section II, we will briefly explain multiresolution analysis of meshes [14], and show its drawbacks in practical implementation, which we extend for irregular triangular meshes. Based on our recent work [19] we consider the inverse problem of wavelet scheme construction and both its connectivity and geometry optimization in section III. Section IV gives comparative results obtained with this new algorithm and a conclusion follows.

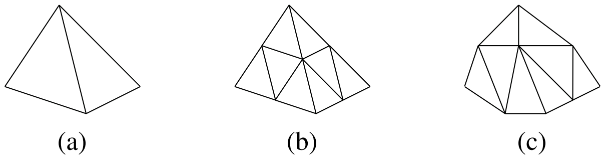


Fig. 1. subdivision-based geometrical refinement : (a) a base mesh is (b) subdivided and (c) its geometry is enhanced to match a finer surface

II. Lounsbery's wavelet based multiresolution scheme

In wavelet decomposition, a mesh (for example a tetrahedron, see figure 1) is quaternary subdivided (figure 1.b) and deformed (figure 1.c), to make it fit the surface to be approximated. This quadrisection is also used for Loop subdivision [10]. These steps can be repeated depending on the required resolution levels.

Multiresolution analysis is computed with two analysis filters A^j and B^j for each resolution level j. The reconstruction (synthesis) is done with two synthesis filters P^j and Q^j. Let us call C^j the $v^j \times 3$ matrix giving the coordinates of each vertex of the mesh M^j having v^j vertices (superscript j is the resolution level). Then we have :

$$C^{j-1} = A^j.C^j \quad (1)$$

$$D^{j-1} = B^j.C^j \quad (2)$$

$$C^j = P^j.C^{j-1} + Q^j.D^{j-1} \quad (3)$$

D^j represents the wavelet coefficients of the mesh, necessary to reconstruct C^j from C^{j-1} . From a theoretical point of view, each column of the P^j matrix (respectively the Q^j matrix) is derived from a scaling function (respectively a wavelet function). These functions are defined on a 3D space fixed by the mesh topology.

To ensure the exact reconstruction of M^j from M^{j-1} and D^{j-1}, the filter-bank must satisfy the following constraint:

$$\left[\frac{A^j}{B^j}\right] = [P^j|Q^j]^{-1} \quad (4)$$

To make the mesh approximation M^{j-1} as close as possible to the original mesh M^j, the lifting scheme [17] is used, which consists in constructing wavelet functions, starting from the hat function (this wavelet is called the "lazy" wavelet), orthogonal

*:Research and Applications Center for Image and Signal Processing, CNRS research unit (UMR 5515). email: {valette,prost}@creatis.insa-lyon.fr

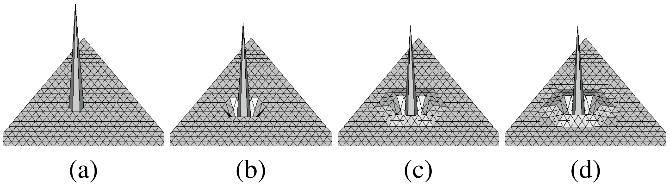


Fig. 2. Different wavelet examples : (a) lazy wavelet, (b) 0-ring support wavelet, (c) 1-ring support wavelet, (d) 2-ring support wavelet.

to the scaling functions which are hat functions too, but with a twice wider support. The main material for the lifting is the inner product between two functions defined by Lounsbery as:

$$< f,g >= \sum_{\tau\in\Delta(M^j)} \frac{K_j}{area(\tau)} \int_\tau f(s)g(s)ds \qquad (5)$$

where $\Delta(M^j)$ is the set of triangles τ of the mesh M^j and K_j is a constant for a given resolution level j $(K_j = 4^{-j)}$. Note that this definition assumes that the triangular faces of the mesh have the same area. As a consequence, the more the area of the mesh faces differs, the less accurate the approximation is. To reduce the computational cost of the algorithm and guarantee its linear complexity, the support of each wavelet function constructed for a particular vertex is restricted to a close neighborhood of the vertex, as depicted in figure 2. The more restricted the support is, the faster the algorithm will be to compute [16]. In this paper, we used wavelets from their lazy version (figure 2.a) to their lifted version where the support is restricted to the 2-ring neighborhood of the considered vertex (figure 2.d).

Wavelets provide a powerful tool for multiresolution analysis of surfaces. However, the major drawback of Lounsbery's scheme is that faces are always merged four by four to construct the approximation. Then the high resolution mesh must have Loop subdivision connectivity. Note that Taubin proposed in [18] an algorithm to find if a given mesh has subdivision connectivity and to reconstruct its approximation by inverse Loop subdivision. If the mesh does not respect this connectivity constraint, one has to process a resampling of the mesh, known as remeshing, which results in a mesh having more faces than the original, as explained in detail in [5] and [9]. The aim of this work is to overcome this difficulty by improving the subdivision process, as described in the next section.

III. A PROPOSAL FOR IRREGULAR MULTIRESOLUTION ANALYSIS

A. Avoiding the remeshing step

The aim of this paper is to provide a new method allowing multiresolution analysis directly on irregular meshes, avoiding the remeshing step, as shown in figure 3. This would result in two major improvements in multiresolution analysis on meshes:

- No extra computation is needed (for the remeshing)
- The reconstruction of the mesh leads to a mesh identical to the original mesh. This allows progressive encoding.

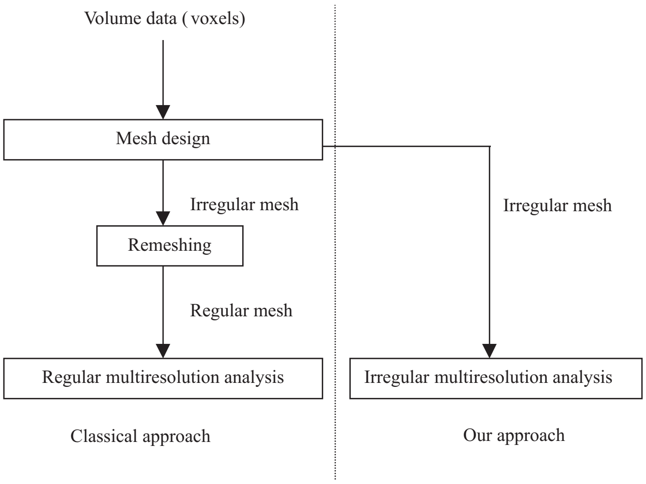


Fig. 3. Regular versus Irregular Wavelet Scheme

Applying the multiresolution scheme on irregular meshes requires the modification of the two main steps:

- The subdivision step, which gives the relationship between the different level meshes (connectivity)
- The analysis-synthesis step, where the vertex coordinates of the lower resolution mesh and the wavelet coefficients are computed (geometry)

These two modifications are described in details in the next two sections.

B. Modelling irregular subdivision scheme is an inverse problem

In the regular multiresolution scheme, the connectivity of all different level meshes depends on the lowest level mesh connectivity. Then the highest resolution level mesh connectivity has to be highly regular. Unfortunately, classically built meshes (for example, meshes built with the marching cubes algorithm [11] or with [12]), are not regular and cannot be directly used. As a result, the subdivision scheme has to be changed, in order to allow every mesh to be processed. Based on our previous work [19], we propose an enhanced subdivision process, where the subdivision differs from one face to another.

In our scheme, each face of a mesh can be subdivided into four, three or two faces, or remain unchanged. Figure 4 depicts the possible cases of subdivision for one face.

1) The direct problem (subdivision): Taking a mesh M^j having n^j faces and v^j vertices, we call S^j a subdivision scheme applied to it, represented by a row vector s^j containing n^j elements (integers between 1 and 11), and describes the subdivision case for each face:

$$s^j = [q_1^j q_2^j \cdots q_{n^j}^j] \qquad (6)$$

where $1 < q_k^j < 11$, according to figure 4. M^{j+1} is then the result of the subdivision process:

$$S^j(M^j) = M^{j+1} \qquad (7)$$

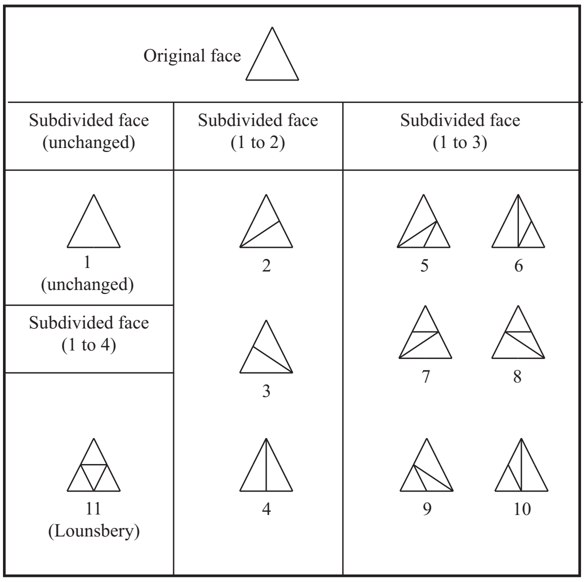
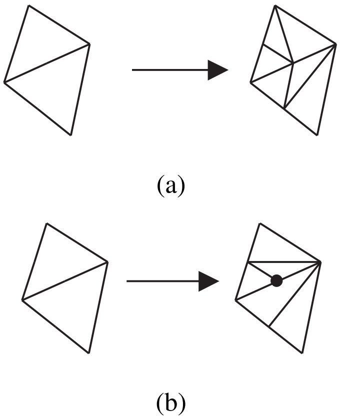


Fig. 4. Possible cases of subdivision


(a)

(b)

Fig. 5. The manifold constraint : (a) manifold result (b) non-manifold result (marked vertex)

We define the merging ratio r^j as:

$$r^j = \frac{n^{j+1}}{n^j} \tag{8}$$

Note that:

$$1 \leq r^j \leq 4 \tag{9}$$

There are 11^{n^j} possible subdivision schemes, but not all of them lead to a manifold mesh, as shown in figure 5. The subdivision of the two faces in figure 5.a results in a manifold mesh, but in figure 5.b, the result is non-manifold.

2) The inverse problem (merging): In order to apply multiresolution analysis by the wavelet decomposition to a given mesh M^j, one can find a mesh M^{j-1} and a subdivision scheme S^{j-1} satisfactory:

$$S^{j-1}(M^{j-1}) = M^j \tag{10}$$

This is a blind inverse problem. For maximum efficiency in compressing the mesh, we try to make the ratio r^{j-1} as near as 4. This consists in merging the faces of the mesh M^j, leading to a mesh having the lowest possible number of faces.

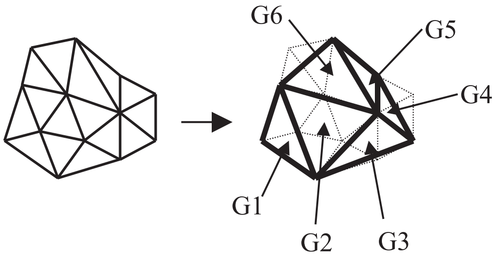
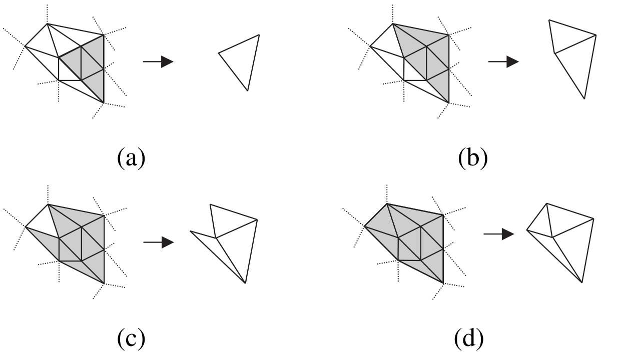


Fig. 6. An example of mesh simplification


(a)

(b)

(c)

(d)

Fig. 7. Expansion of the simplified face set

Figure 6 shows an example, where 15 faces are reduced to 6, resulting from merging 4:1 faces for G2, 3:1 faces for G3 and G6, 2:1 faces for G1 and G4 and keeping one face unchanged for G5. For this subdivision scheme, $r^{j-1} = \frac{15}{6} = 2.5$. Briefly, the simplification algorithm starts by selecting four faces, building a set of merged faces, and tries to expand this set by merging faces around it. Figure 7 shows the beginning of the expansion of the merged faces set (in gray), merging sequentially, 2:1 faces, 3:1 faces, and leaving one face unchanged.

During the simplified face set expansion, visited vertices are labeled as parent or child vertices. As the simplified face set grows over the input mesh, its boundaries may encounter each other, and the algorithm sometimes faces cases where it cannot merge some faces as wanted, because the faces don't match any expected irregular subdivision case. In these cases, a modification of the mesh is allowed. It consists in an edge flip between two neighbor faces, as shown in figure 8. Of course this modification information has to be stored, to recover the original mesh after subdivision and guarantee the reversibility of the simplification process.

We notice that this modification will introduce a quality loss in terms of mean squares error of the approximation of M^j with M^{j-1}. But the difference between the original mesh and the modified mesh is small and experimental results show that this quality loss can be ignored. The next part describes the algorithm in detail.

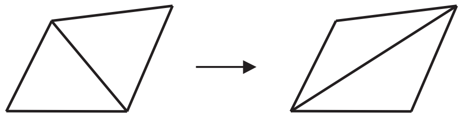


Fig. 8. An edge flip for two adjacent faces

C. The merging algorithm

First, we have to define three codebooks, describing the different merge-split cases involved in the proposed algorithm:

- The Wavelet Codebook $W = \{p^u_t\}$. Figure 9.a describes the subdivision cases (1:1, 1:2, 1:3, 1:4). Superscript u denotes the order of subdivision (1:u). Note that u is also the number of faces in the patches.
- The Incident Codebook $I = \{f^v_e\}$ Figure 9.b depicts all the cases of allowed edge flips combined with subdivisions. Superscript v denotes the number of faces for each patch.
- The Merge Codebook $M = \{g^w_l\}$. Figure 9.c shows how triangles are merged : each patch in the Wavelet Codebook or in the Incident Codebook has an associated merge patch in the merge Codebook. Note that every patch in the Wavelet Codebook is associated to g^1_1.

In these three codebooks, the white marked vertices are called parent vertices, and the black marked ones are called child vertices. Child vertices disappear during the mesh simplification, while the parent vertices are kept. One child vertex has always two associated parent vertices. Note that some constraints must be respected during the mesh simplification step, in order to keep this step reversible:

- A vertex can be labeled as a child vertex only if its valence is equal to 4, 5 or 6.
- Some child vertices can share a parent vertex, but two vertices must never have the same parents (figure 10.a).
- A vertex V_1 cannot have vertices V_2 and V_3 as parent vertices if the edge V_2V_3 exists in the mesh (figure 10.b).
- An edge flip cannot occur if the resulting created edge already exists (figure 10.c).

With this material, one can build a reversible algorithm that can simplify a given mesh M^j to a lower resolution one M^{j-1}.

Next is the description of the merging algorithm. First, we note as "matching" both the full search for the best match in the corresponding codebook and a test on the number of edges outside the merged region for each vertex to be removed. This number of outside edges must be one or two to prevent the algorithm from being stuck in a trap configuration (see example in figure 10.c). Then the merging algorithm consists of the following operations :

- Choose a seed triangle s. The seed triangle and its three neighbor triangles s_1, s_2 and s_3 define the simplified face set $F_0 = \{s, s_1, s_2, s_3\}$ and the to-be-simplified face set $\overline{F_0}$ with $F_0 \bigcup \overline{F_0} = M^j$. It also defines the boundary edge set $E_0 = \{e_{0,k_{k=0\ldots N_0}}\}$ as $E_0 = F_0 \bigcap \overline{F_0}$. The integer N_0 is the number of edges on the boundary of F_0 minus 1 (as an example, $N_0 = 5$)

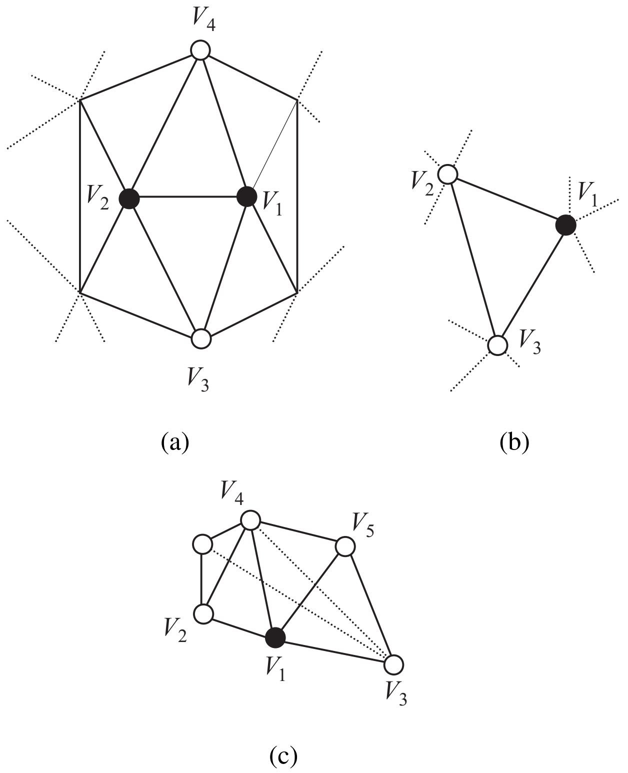


Fig. 10. Forbidden simplification cases: (a) two vertices V_1 and V_2 have the same parent vertices V_3 and V_4; (b) the vertex V_1 has V_2 and V_3 as parent vertices, but the edge V_1V_2 exists; (c) the edge flip that would change V_1V_5 to V_3V_4 is not allowed as V_3V_4 already exists in the mesh. On the other hand, flipping V_1V_4 to V_2V_5 is allowed

- Set $i = 0$
- N_i is the number of edges on the boundary of F_i minus 1
- While $N_i \geq 0$ do
  - For k=0 to $N_i - 1$:
    - select in the Wavelet Codebook $p^u_m = (p^v_n$ matching $\overline{F_i}$ at $e_{i,k})/u = max(v)$
    - if no match found : select in the Incident Codebook $f^w_l = (f^v_n$ matching $\overline{F_i}$ at $e_{i,k})/w = max(v)$
  - $F_{i+1} = \bigcup_m$selected $p^u_m \bigcup_l$selected $f^w_l \bigcup F_i$
  - Create M^{j-1}_{i+1} by merging each selected p^u_m and each f^w_l according to its associated patch in the Merge Codebook $M = \{g^l_w\}$
  - Set $i = i + 1$
  - Define the to-be-simplified face set $\overline{F_i}$ and the boundary edge set $E_i = \{e_{i,k}\}_{k=0\ldots N_i}$ such that $F_i \bigcup \overline{F_i} = M^j$ and $E_i = F_i \bigcap \overline{F_i}$
- $M^{j-1} = M^{j-1}_i$ is then the new simplified mesh at resolution level $j - 1$

Figure 11 shows an example of the progression of the proposed algorithm for a small triangular face set:

- The chosen seed triangle is shown in figure 11.a. $F_0 = \{s, s_1, s_2, s_3\}$, and $E_0 = \{e_0, e_1, e_2, e_3, e_4, e_5\}$ consists of the six edges surrounding F_0.
- Figures 11.b and 11.c show the patterns matching $\overline{F_0}$ at

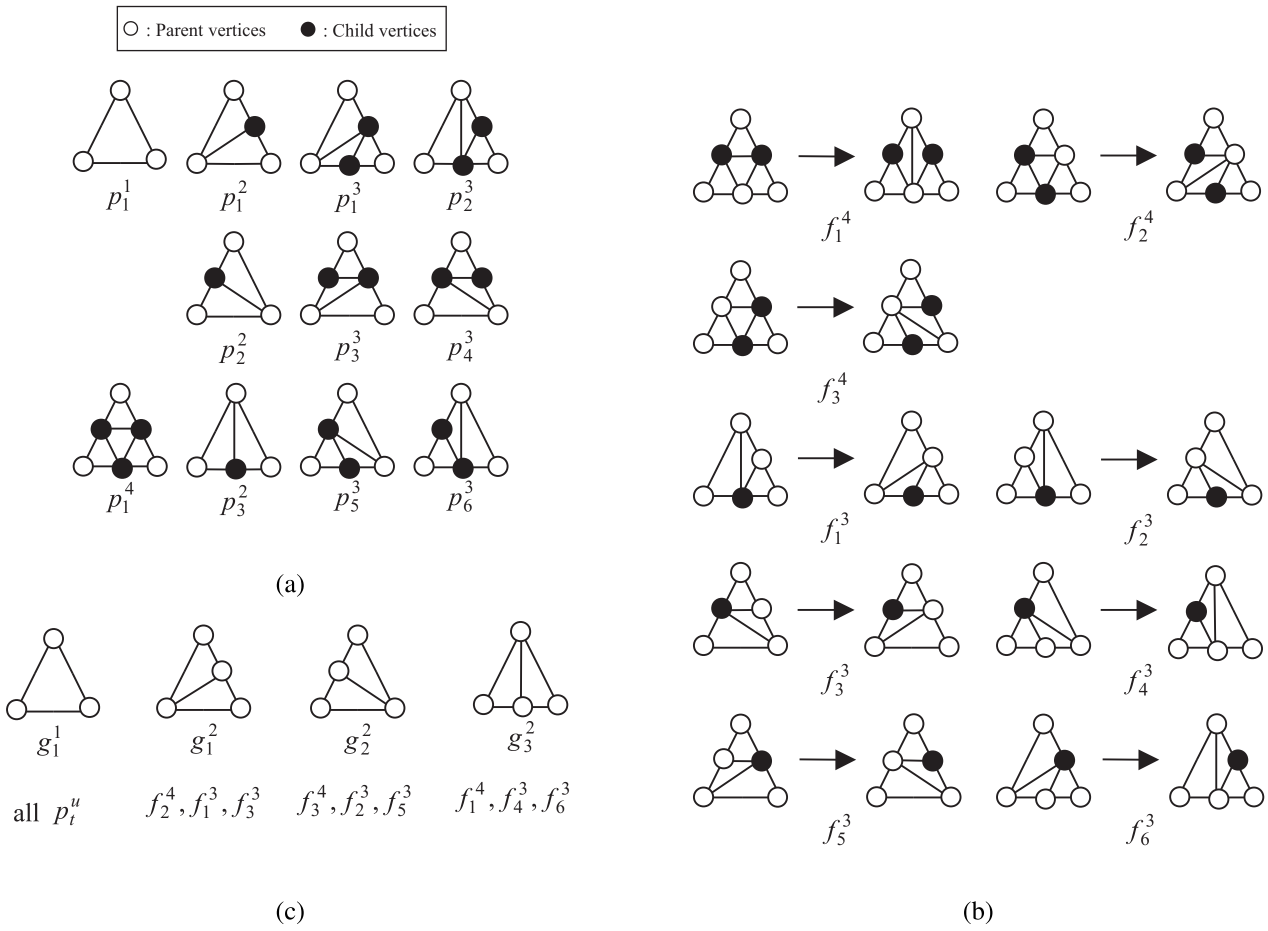


Fig. 9. Codebooks : (a) Wavelet Codebook $W = \{p_t^u\}$; (b) Incident Codebook $I = \{f_e^v\}$; (c) Merge Codebook $M = \{g_l^w\}$

$e_{0,0}$ and $e_{0,1}$. Finally, the chosen patch for these edges is p_2^3, for optimality.

- figure 11.d depicts the algorithm configuration after the first loop. Three wavelet patches were selected: p_2^3 for $e_{0,0}$ and $e_{0,1}$, f_3^3 for $e_{0,2}$ and $e_{0,3}$, p_3^2 for $e_{0,4}$ and $e_{0,5}$. The simplified face set F_1 is the gray colored region. In figure 11.e the resulting merged mesh M_{j-1}^1 is shown.
- figure 11.f : after the second loop, F_2 covers the entire face set except one face. Four new patches have been selected.
- figure 11.g : F_3 covers the entire face set and the simplification step is complete. The final simplified mesh M_{j-1} is shown in figure 11.h.

In this example, 20 triangles were merged into 10 triangles. Finally, the algorithm is very efficient for simplifying meshes. So far, we have not seen a mesh that we could not simplify efficiently. Note that the number of resolution levels (merging efficiency) depends on the choice of the seed triangle. However, in practice, we have observed only minor efficiency variations when choosing another seed triangle. Also, if the original mesh is a quadrisected one (i.e. it results from Loop subdivision), our algorithm finds the original base mesh in 25% of the cases. For 100% efficiency with quadrisected meshes as in [18], our algorithm would need to run four times with four different adjacent seed triangles[1].

D. Optimizations

Although our algorithm is able to simplify every mesh we had in our possession, we improved it by introducing optimization constraints, leading to better results.

1) Connectivity-based optimization: For a given high resolution mesh M^j, the optimal simplification would be obtained if all the faces of M^j were merged four by four. This happens only when M^j has a subdivision connectivity, where all child vertices have their valence equal to 6. Applying a new simplification step to M^{j-1} would lead to the optimal simplification M^{j-2} if M^{j-1} had a subdivision connectivity, where all child vertices again have their valence equal to 6. This recurrent demonstration leads us to the conclusion that optimal results are obtained when the input mesh has vertices with a valence equal to 6 most of the time. Unfortunately, natural meshes do not always have this kind of regularity, but our algorithm can be driven so that the approximation meshes are as regular as possible. Looking at the merge and incident codebooks defined in section 3, we can see that the valence of some parent vertices is changed after the simplification step :

[1] Note that a solution to obtain directly the original base mesh is proposed in [21]

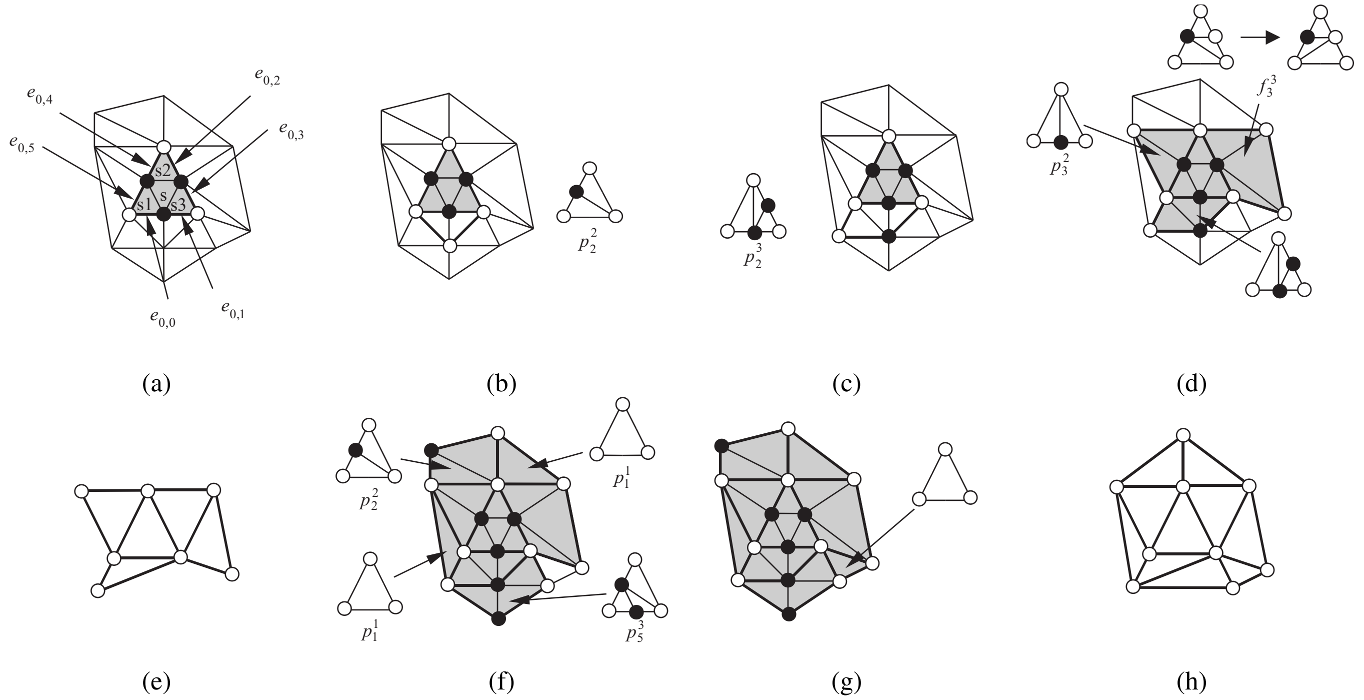


Fig. 11. Example of iterative mesh simplification

- for each three-to-one and two-to-one merging patch, the valence of a parent vertex is decreased by one.
- for each edge flip, two vertices have their valence decreased by one, and two vertices have their valence increased by one.

As a consequence, when several patches match a given edge during the simplification step, we chose the patch that changes the valence of the parent vertices to the more regular local configuration. Figure 12 gives an example of such optimization: two patches match the edge V_1V_2, the first one (figure 12.a) leads to the creation of the face $V_1V_2V_4$ in the simplified mesh. The second patch (figure 12.b) leads to the creation of the face $V_1V_2V_3$. In the first case, V_1 and V_2 have their valence changed to 6 for both vertices. In the second case their valence is also changed to 5 for V_1 and 7 for V_2. As a consequence, the first patch is chosen, so that the vertices have their valence well balanced. With this connectivity-based optimization, the mesh regularity increases during its simplification, and the proposed scheme is considerably improved.

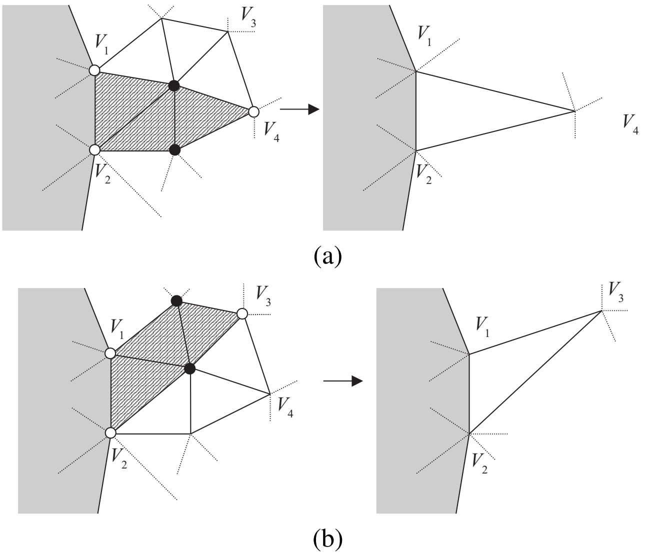


Fig. 12. Connectivity-based optimization : the simplified face set is the gray colored area. 2 three-to-one patches can be chosen to expand it. (a) V_1, V_2 and V_4 are chosen to construct a new face in the simplified mesh. (b) V_1, V_2 and V_3 are chosen. The best choice between these two patches is the first one, as it leads to valences equal to 6 for both vertices V_1 and V_2 instead of valence 5 and 7 with the second choice

2) Geometry-based optimization: At this point in the paper, the geometry of the input mesh M^j is not taken into account. Actually, the simplification is only based on the connectivity of M^j to construct the connectivity M^{j-1}. Afterwards, the geometry of M^{j-1} is obtained by approximating M^j with the proposed irregular wavelet decomposition. The wavelet scheme is driven by the hierarchical relationship between the connectivity of M^j and the connectivity of M^{j-1}, and ensures the best possible approximation with the given hierarchical dependency. Instead of the previous geometry-blind simplification scheme, the approximation can be improved by selecting which vertices to remove (the child vertices) and which vertices to associate with them (their parent vertices) in accordance with a geometric criterion. We propose here a two stage geometry-based criterion, that we call Wavelet Geometrical Criterion (WGC), embedded in the inverse problem solver as an additional constraint. When the solver tries to expand the simplified face set, it labels vertices as parent vertices or child vertices (with two associated parents). The constraint appears when testing if a vertex V_1 can be labelled as child vertex, with V_2 and V_3 as parent vertices :

- We first verify if this labelling satisfies the connectivity

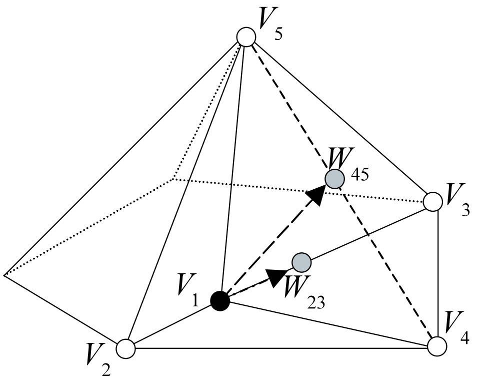


Fig. 13. Geometrical criterion for mesh simplification: V_1 is a sharp vertex and cannot be removed properly if it has V_4 and V_5 as parent vertices. On the other hand, it can be removed if its parent vertices are V_2 and V_3, since this removal won't change the geometry significantly.

constraints defined in section III.C. If these constraints are not respected, then the request is rejected.

- If the connectivity constraints are well respected, then the geometry of the mesh is taken into account. The first geometrical criterion is the sharpness of V_1: if V_1 is not a sharp vertex it can be labelled as child vertex without any supplementary test, since its removal won't change the mesh geometry very much. We define V_1 as a sharp vertex if any dihedral angle between two neighbor faces around V_1 is superior to a given threshold t_{sharp}.
- If V_1 has been detected as a sharp vertex, one more test has to be done with the proposed parent vertices V_2 and V_3. We define the wavelet ratio R_w as:

$$R_w = \frac{\|\overrightarrow{V_1V_2} \wedge \overrightarrow{V_2V_3}\|}{\overrightarrow{V_2V_3}^2} = \frac{\|\overrightarrow{V_1W} \wedge \overrightarrow{V_2V_3}\|}{\overrightarrow{V_2V_3}^2} \tag{11}$$

where W is the midpoint of the edge V_2V_3. If R_w is superior to a given threshold $T_{Wavelet}$, then V_1 cannot be labeled as a child vertex with V_2 and V_3 for parent vertices. Note that even if the test is negative, V_1 may be further labeled as a child vertex with other parents than the pair (V_2, V_3).

An example is shown in figure 13: we look forward to removing V_1, which is a sharp vertex. Its parents can be either V_4 and V_5 or V_2 and V_3. For the first solution, R_w mainly depends on the product $\overrightarrow{V_1W_{45}} \wedge \overrightarrow{V_4V_5}$ which is non negligible as $\overrightarrow{V_1W_{45}}$ and $\overrightarrow{V_4V_5}$ are far from collinear. Then V_1 cannot be labeled as a child vertex with V_4 and V_5 as parent vertices. For the second solution, R_w depends on the product $\overrightarrow{V_1W_{23}} \wedge \overrightarrow{V_2V_3}$ which is almost zero as $\overrightarrow{V_1W_{23}}$ and $\overrightarrow{V_2V_3}$ are almost collinear.

T_{sharp} and $T_{wavelet}$ are the two tuning parameters which drive the geometric constraint during the simplification step. Note that setting these thresholds too low could prevent the merging algorithm from simplifying the mesh. However, in our different tests, we used $T_{sharp} = 0.3\,rad$ and $T_{wavelet} = 0.2$, without any lockup. For the fandisk mesh, we set $T_{wavelet} = 0.2$, which is a more selective threshold, because this mesh is mainly made of perfectly flat surfaces and very sharp edges. Figure 14 shows the different results obtained on the "fandisk" mesh : the sharp edges sometimes disappear with the connectivity-only based simplification algorithm (figure 14.b). With WGC, the shape resulting simplified mesh (figure 14.c) is much closer to the original mesh (figure 14.a).

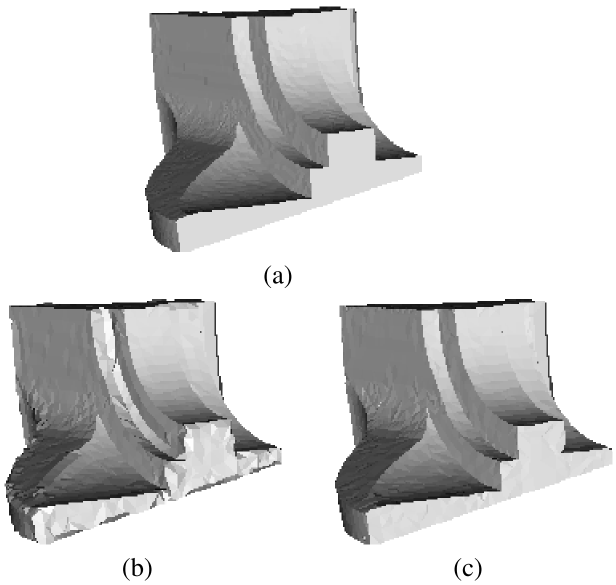


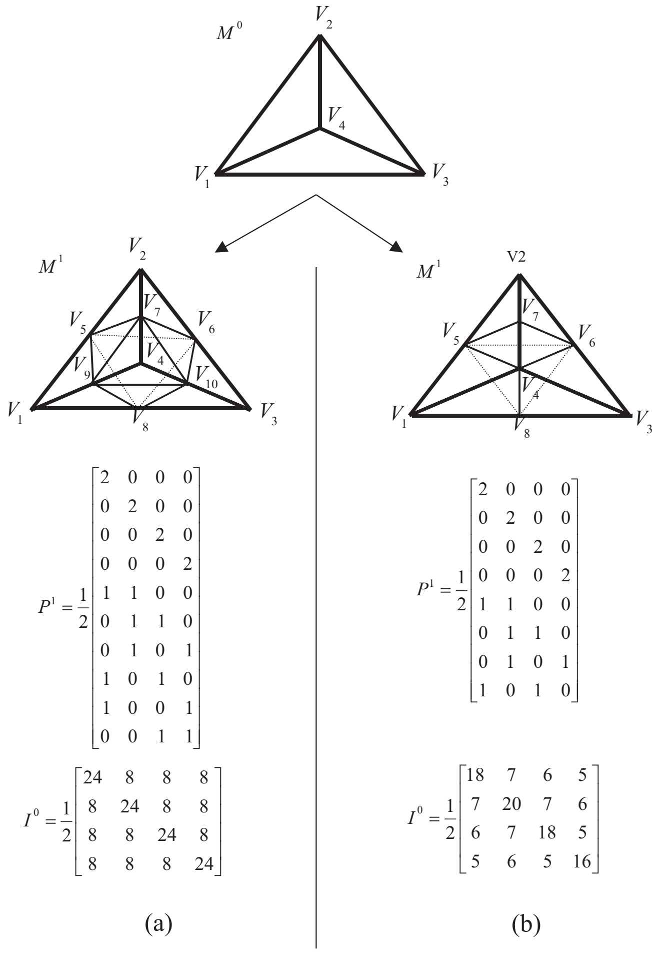
Fig. 14. Approximations of the "fandisk" mesh: (a) original mesh (6475 vertices) , (b) mesh simplification without geometry criterion (2266 vertices) : some sharp edges were removed, (c) mesh simplification with a geometric criterion (2438 vertices): sharp edges are well preserved, keeping the visual aspect of the approximation close to the original mesh.

E. An inner product for filter-bank construction

Once the simplified mesh connectivity has been constructed, we have to compute its geometry, that is to calculate the analysis filters A^j and B^j from which we can compute the simplified mesh vertex coordinates. In sharp contrast with Lounsbery's scheme, due to the change of the subdivision process, the inner product (5) has to be reformulated and becomes:

$$< f, g >= \sum_{\tau \in \Delta(M^j)} \frac{K_j(\tau)}{area(\tau)} \int_\tau f(s)g(s)ds \tag{12}$$

where $K_j(\tau)$ is no longer a constant for a given resolution level and changes with each face of the mesh. As an example, a face split into 3 faces will have $K_j(\tau) = 3$ and the three resulting faces will have $K_{j+1}(\tau) = 1$, taking into account the differences between the triangle areas : the area of the first face cited above will be approximately three times wider than the last three. With this new inner product definition, Lounsbery's scheme can be extended to the calculation of the analysis-synthesis filters as described in section II. Figure 15 briefly compares both approaches : a tetrahedron is regularly subdivided in figure 15.a and irregularly subdivided in figure 15.b. The synthesis filter P^1 and the inner product matrix I^0 are shown for both cases. We can see that both approaches lead to similar matrices, the main difference being between the inner product matrices.



$$P^1=\frac{1}{2}\begin{bmatrix}2&0&0&0\\0&2&0&0\\0&0&2&0\\0&0&0&2\\1&1&0&0\\0&1&1&0\\0&1&0&1\\1&0&1&0\\1&0&0&1\\0&0&1&1\end{bmatrix}$$

$$I^0=\frac{1}{2}\begin{bmatrix}24&8&8&8\\8&24&8&8\\8&8&24&8\\8&8&8&24\end{bmatrix}$$

(a)

$$P^1=\frac{1}{2}\begin{bmatrix}2&0&0&0\\0&2&0&0\\0&0&2&0\\0&0&0&2\\1&1&0&0\\0&1&1&0\\0&1&0&1\\1&0&1&0\end{bmatrix}$$

$$I^0=\frac{1}{2}\begin{bmatrix}18&7&6&5\\7&20&7&6\\6&7&18&5\\5&6&5&16\end{bmatrix}$$

(b)

Fig. 15. Regular vs irregular subdivision of a tetrahedron : (a) regular case where all the faces are quadrisected. (b) irregular case where a face is split into four faces, two faces are split into three faces and one face is split in two

IV. Results

This section shows experimental results obtained by our implementation. Our algorithm was able to build approximations for every mesh we tested on it. For any mesh of genus 0 without boundaries, we were able to simplify it to a tetrahedron. For other kinds of meshes, the size of simplest approximation depends on the input mesh. As an example, the Stanford Bunny, which has 5 holes, was reduced to a mesh with 22 vertices, and the ”eight” mesh which has genus 2 was reduced to a mesh with 15 vertices. Figure 16 shows some approximation levels obtained with the ”fandisk” mesh (6475 vertices), with 2438, 1250, 648, 378, 230, 155 and 83 vertices, respectively. Figure 17 shows some results obtained on ”bunny”, ”blob” and ”eight” : (a) is the original mesh, (b) is the first approximation and (c) shows a mesh having about 10 times fewer vertices than the original. We can see that the approximation remains good in low levels.

Table I compares the mean square error computed by the Metro tool [3] for the first approximation level, with both different configurations and several reference models. For each mesh, we tried to change the size of the wavelet support, to use a ”connectivity-only” simplification scheme or to use WGC. Computation time is also given for the construction of all the approximation levels of the ”bunny mesh” (not only the first one), using an Intel Pentium III PC running at 500Mhz. The last two rows show the average error for the collection, and the approximation error decrease over using our algorithm without the lifting scheme and without any geometric criterion. These figures show that using the lifting scheme (0-ring wavelet support) decreases the error by more than 30%, and that using the lifting for a larger wavelet support does not improve the results significantly (32.1% for a 2-ring wavelet support vs 31.7% for a 0-ring wavelet support), while it decreases significantly the speed of the algorithm (40.2 seconds vs 14.5 seconds). Using WGC decreases the error by 19.5%, and combining the lifting scheme with WGC during simplification improves the quality by 42.4%, if we use a 0-ring or a 2-ring sized wavelet support. As a conclusion, the most effective combination is the combination of the lifting scheme for a 0-ring wavelet support WGC, for a fast and very effective multiresolution analysis of irregular meshes.

V. Conclusion

We proposed the enhancement of the new scheme [19] for multiresolution analysis on arbitrary meshes. In sharp contrast with [5] and [9] where a resampling of the original mesh is necessary, our scheme processes directly on the original mesh. The irregular multiresolution scheme is an inverse problem for which we proposed an efficient solution based on the connectivity and the geometry of the input mesh. The proposed method has many potential applications such as mesh compression and progressive transmission [20], fast rendering of 3D images, acceleration for volume extraction and segmentation [13] and multiresolution mesh processing algorithms as in [6].

Acknowledgments

This work is in the scope of the scientific topics of the GdR-PRC ISIS research group of the French National Center for Scientific research.

The authors would like to thank the reviewers both for their helpful comments and suggestions. Their criticisms helped the authors to improve the quality of this paper.

References

[1] Pierre Alliez and Mathieu Desbrun, “Progressive encoding for lossless transmission of 3d meshes,” in *ACM Siggraph Conference Proceedings*, 2001, pp. 198–205.

[2] G.-P. Bonneau, “Multiresolution Analysis on Irregular Surface Meshes,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 4(4), 1998.

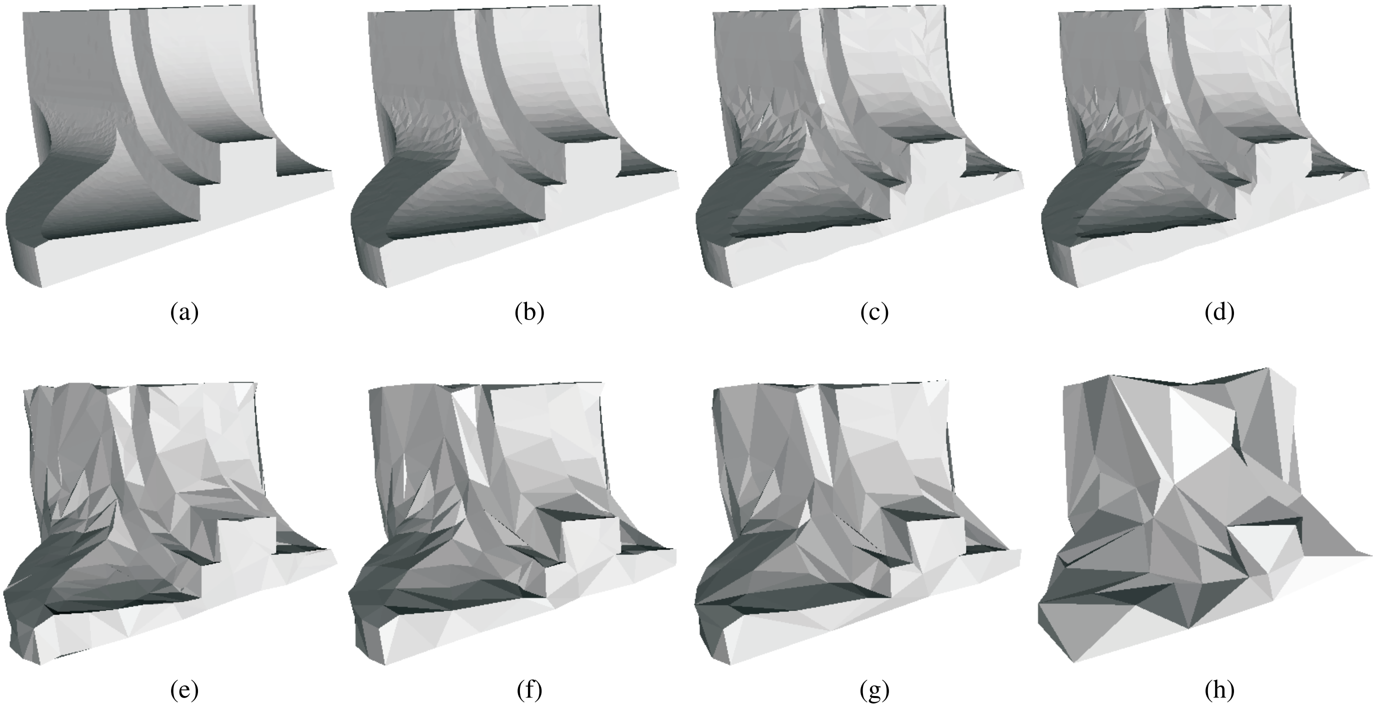
[3] P. Cignoni, C. Rochini, and R. Scopigno, “Metro : Measuring error on simplified surfaces,” *Computer Graphics Forum*, , no. 17(2), pp. 167–174, 1998.

[4] D. Cohen-Or, D. Levin, and O. Remez, “Progressive compression of arbitrary triangular meshes,” in *IEEE Visualization 99*, 1999, pp. 67–72.

[5] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, “Multiresolution analysis of arbitrary meshes,” in *ACM Siggraph Conference Proceedings*, 1995, pp. 173–182.

[6] I. Guskov, W. Sweldens, and P. Schröder, “Multiresolution signal processing for meshes,” *ACM Siggraph Conference Proceedings*, 1999.

[7] P. S. Heckbert and M. Garland, “Survey of polygonal surface simplification algorithms,” 1997, ACM Siggraph conference course notes.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 16. Results on the Fandisk : (a) original mesh (6475 vertices), (b) to (h) : successive approximations meshes of 2438, 1250, 648, 378, 230, 155 and 83 vertices, respectively

TABLE I

METRO MEAN SQUARE ERROR (% OF BOUNDING BOX) FOR THE FIRST APPROXIMATION LEVEL

Geometric criterion	No				Yes		
Wavelet support	Without lifting	0-ring	1-ring	2-ring	Without lifting	0-ring	2-ring
Blob	0.1197	0.0947	0.0936	0.0937	0.1061	0.0876	0.0873
Bunny	0.0309	0.0253	0.0249	0.0249	0.0287	0.0239	0.0236
(time in seconds)	(11.7)	(14.5)	(19.8)	(40.2)	(13.6)	(16.4)	(41.4)
Eight	0.3204	0.1828	0.1834	0.1816	0.3204	0.1828	0.1816
Fandisk	0.1337	0.0992	0.0992	0.0996	0.0248	0.0348	0.0420
Horse	0.0305	0.0252	0.0250	0.0250	0.0269	0.0228	0.0229
Mannequin	0.0320	0.0238	0.0236	0.0236	0.0261	0.0257	0.0205
Venus Head	0.0478	0.0377	0.0374	0.0373	0.0423	0.0341	0.0340
Average	0.102	0.0698	0.0696	0.0694	0.0822	0.0588	0.0588
Improvement	0	31.7%	31.9%	32.1%	19.5%	42.4%	42.4%

[8] H. Hoppe, “Progressive meshes,” in *ACM Siggraph 96 Conference Proceedings*, 1996, pp. 99–108.

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

[10] C. Loop, Ed., *Smooth subdivision surfaces based on triangles*, Masters thesis. Dept. of Mathematics, University of Utah, 1987.

[11] W.E. Lorensen and H.E. Cline, “Marching cubes : A high rresolution 3d surface construction algorithm,” in *ACM Siggraph Conference Proceedings*, 1987, pp. 163–169.

[12] J. Lotjonen, P.J. Reissman, I.E. Magnin, J. Nenonen, and T. Katila, “A triangulation method of an arbitrary point set for biomagnetic problem,” *IEEE Transactions on Magnetics*, vol. 34(4), pp. 2228–2233, 1998.

[13] J. Lotjonen, P.J. Reissman, I.E. Magnin, and T. Katila, “Model extraction from magnetic resonance volume data using the deformable pyramid,” *Medical Image Analysis*, vol. 3(4), pp. 387–406, 1999.

[14] M. Lounsbery, *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*, PhD thesis. Dept. of Computer Science, University of Washington, 1994.

[15] M. Soucy and D. Laurendeau, “Multiresolution surface modeling based on hierarchical triangulation,” *Computer vision and image understanding*, vol. 63(1), pp. 1–14, 1996.

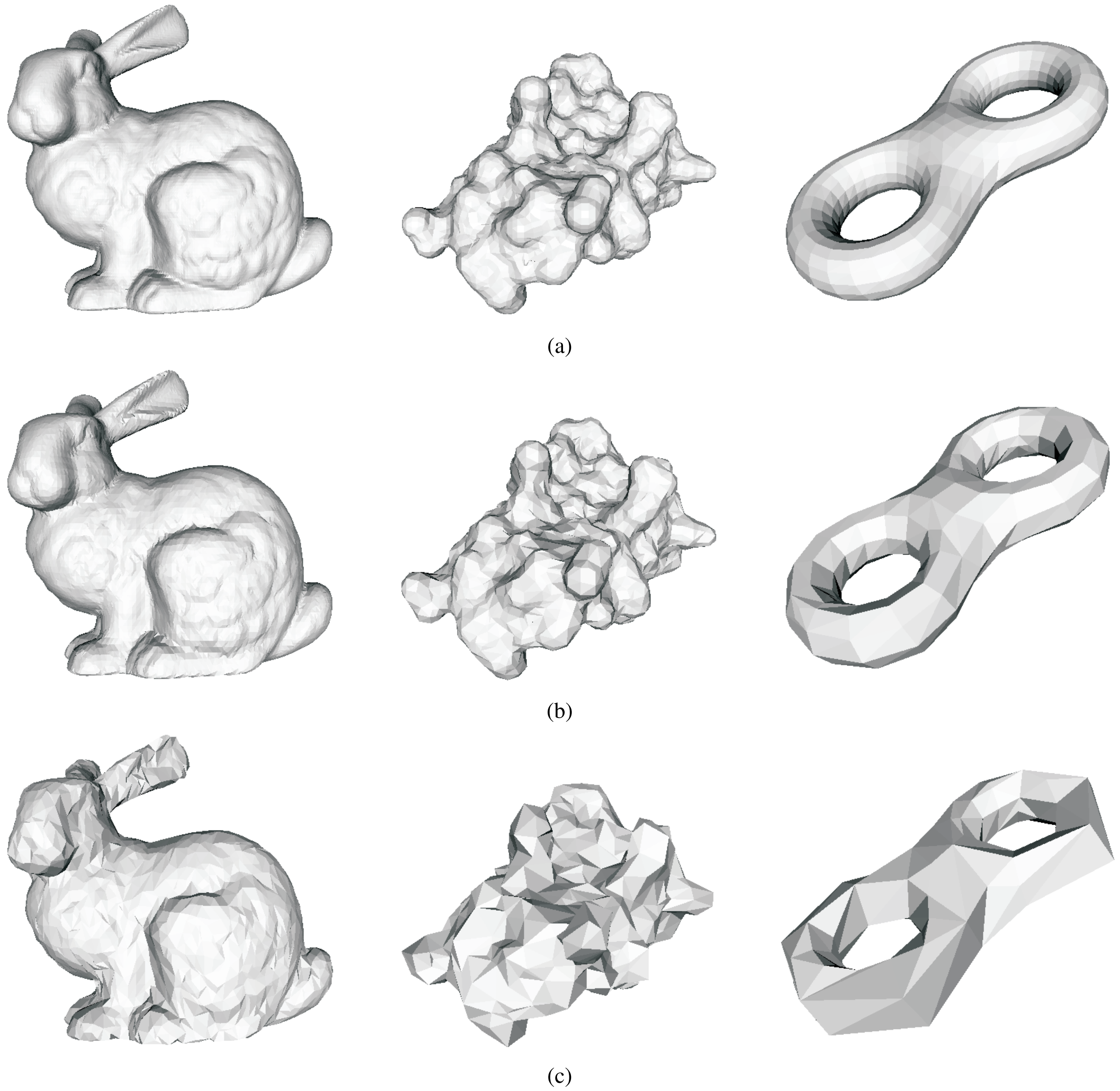
[16] E. Stollnitz, T. DeRose, and D. Salesin, *Wavelets for Computer Graphics*, Morgan Kaufmann Publishers, 1996.

[17] W. Sweldens, “The lifting scheme : A custom-design construction of biorthogonal wavelets,” *Applied and Computational Harmonic Analysis*, vol. 3, pp. 186–200, April 1996.

[18] G. Taubin, “Detecting and reconstructing subdivision connectivity,” *The Visual Computer, Special Issue on Subdivision*, 2001.

[19] S. Valette, Y. S. Kim, H. J. Jung, I. Magnin, and R. Prost, “A multiresolution wavelet scheme for irregularly subdivided 3d triangular mesh,” in *IEEE Int. Conf on Image Processing ICIP99*, October 1999, vol. 1, pp. 171–174.

[20] S. Valette and R. Prost, “A wavelet-based progressive compression



(a)

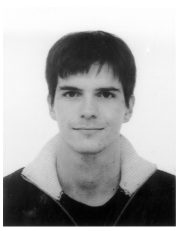
(b)

(c)

Fig. 17. Approximations examples on the Bunny, Blob and Eight meshes : (a) original meshes (b) first approximations (c) approximations with meshes with about 10 times fewer vertices than the original models

scheme for triangle meshes : Wavemesh," *IEEE Transactions on Visualization and Computer Graphics, to appear, available upon request*, 2003.

[21] S. Valette, J. Rossignac, and R. Prost, "An Efficient Subdivision Inversion for Wavemesh-Based Progressive Compression of 3D Triangle Meshes," in *IEEE Int. Conf on Image Processing ICIP03*, 2003.



Sébastien Valette Was born in France, in 1975. He recieved the M.S. Degree from the Electrical Engineering Department, at the National Institute for Applied Sciences (INSA) of Lyon, France, in 1998. He obtained the PhD Degree at INSA of Lyon in 2002. His research interests include 3D processing, wavelets, progressive compression and multiresolution analysis.

Rémy Prost received his doctorate degree in Electronics Engineering and his ”Docteur es Sciences” degree from Lyon University and the National Institute of Applied Sciences (INSA), Lyon, France, in 1977 and 1987 respectively. He is currently a professor in the Department of Electrical Engineering at INSA -Lyon. Both his teaching and research interests include digital signal processing, inverse problems, image data compression, multiresolution algorithms, wavelets, and meshes processing. He leads the ’ Volume (3D) Image Processing ’ project in the CREATIS Laboratory (CNRS #5515) at INSA-Lyon. Since 1982 he is a member of the IEEE.