

# Volume Preservation of Multiresolution Meshes

Basile Sauvage\*  
University of Strasbourg

Stefanie Hahmann†  
University of Grenoble (INPG)

Georges-Pierre Bonneau‡  
University of Grenoble (UJF)

## Abstract

Geometric constraints have proved to be efficient for enhancing the realism of shape animation. The present paper addresses the computation and the preservation of the volume enclosed by multiresolution meshes. A wavelet based representation allows the mesh to be handled at any level of resolution. The key contribution is the calculation of the volume as a trilinear form with respect to the multiresolution coefficients. Efficiency is reached thanks to the pre-processing of a sparse 3D data structure involving the transposition of the filters while represented as a lifting scheme. A versatile and interactive method for preserving the volume during a deformation process is then proposed. It is based on a quadratic minimization subject to a linearization of the volume constraint. A closed form of the solution is derived.

## 1 Introduction

Wavelet based multiresolution (MR) analysis is now established as an efficient tool for modeling and handling curves and surfaces [6, 22]. It provides an efficient representation of complex functions at multiple levels of detail. Thus convenient handling of geometrical objects is possible. It has been extended to semi-regular meshes, and successfully used for level-of-detail visualization [4, 21] and deformation [17]. It has been combined with a local frame encoding and used for mesh compression [13, 19]. Compared to other multiresolution models for meshes [27, 14, 10], the wavelet approach takes advantage of involving only linear filters that are not geometry dependent. Moreover it is also suitable for smooth surfaces.

Besides, geometric constraints have proved to be useful for deforming curves and surfaces. Linear constraints such as position, tangency, orthogonality, and symmetry [7, 8] are generally related to direct shape manipulation. These constraints offer the advantage of efficient processing, allowing for interactive manipulation of the free form geometry. Satisfying non-linear constraints, in contrast, requires intense computational effort, so that their use for interactive shape manipulation is generally very limited. The volume constraint has been considered for free-form solids [20], FFD [1, 12], vector-field based deformation [25] and meshes [3, 16]. When manipulating surfaces, the volume is a global property because it is given by an integral over the surface. Thus one of the main issues is to get a meaningful local preservation of the volume. Botsch and Kobbelt [3] achieve this by building a MR representation based on the encoding of volume elements between levels. The main drawbacks are expensive optimizations, and a very specific representation. Lipman *et al.* [16] link an approximation of the volume to the

curvature of the surface. It is combined with an intrinsic representation of meshes based on the first and second fundamental forms. By way of compensation the handling of the mesh is not very intuitive.

In the context of multiresolution, editing with area constraints has been studied for planar curves [5, 11]. Both methods are designed for real-time deformations with up to a few thousand control points. Yet, a generalization to the volume of MR meshes is not obvious because of the growth of the complexity, and because of a complex topology.

This paper makes two main contributions. First, it provides formulas and an efficient algorithm to compute the volume enclosed by a mesh with respect to a multiresolution basis. The volume of the fine mesh is expressed through all levels of resolution as a trilinear form of the coarse and detail coefficients. It involves the transposition of the MR filters while they are represented as a lifting scheme made up of local masks. Though it is illustrated with a specific scheme, it works for any multiresolution scheme, as long as a lifting scheme representation is available. Second, an algorithm is provided for real-time deformation of the mesh at any MR level while preserving the volume. It is versatile since it doesn't depend on a particular mesh deformation technique: given a target shape, a correction is performed to meet a reference volume. The correction is computed by linearizing the volume constraint and by solving a constrained minimization problem. Efficiency is achieved thanks to a closed form of the solution.

The paper is organized as follows. Section 2 reminds the basics about semi-regular MR meshes and the lifting scheme. In Section 3 we provide a computation of the enclosed volume, including algorithms and data structures. In Section 4 we derive a volume preserving deformation algorithm, whose efficiency is illustrated by a few examples in Section 5. A conclusion is given in Section 6, together with existing and future extensions of this work.

## 2 Multiresolution semi-regular meshes

The chosen multiresolution analysis is induced by wavelet basis, which are designed for smooth surfaces. However we are mainly interested in the meshes, that correspond to the so-called control meshes in the continuous framework. Thus, for the purpose of concision, we remind here for the principles of MR on the meshes only. The extension to smooth surfaces is discussed in section 6.

Let  $\mathcal{M}^n$  be a semi-regular triangular mesh, *i.e.* issuing from  $n$  primal subdivision steps of a coarse mesh  $\mathcal{M}^0$ . Let  $N$  be the number of vertices of  $\mathcal{M}^n$ . Its embedding  $\mathbf{c}^n(\mathcal{M}^n)$  maps any vertex  $i$  of  $\mathcal{M}^n$  onto a position  $\mathbf{c}_i^n \in \mathbb{R}^3$ ,  $i = 1, \dots, N$ . Then the filterbank algorithm [18] defines the analysis and synthesis processes (see Figure 1): From left to right, the fine mesh is analyzed (or decomposed) into an approximating coarse mesh and sequences of details; From right to left, the synthesis computes the inverse transformation. From the geometry  $\mathbf{c}^l$  of an intermediate mesh  $\mathcal{M}^l$ ,  $1 \leq l \leq n$ , an *analysis step* consists in computing both i) the geometry  $\mathbf{c}^{l-1} = A^l \mathbf{c}^l$  of a coarser mesh  $\mathcal{M}^{l-1}$  and ii) some details coefficients  $\mathbf{d}^{l-1} = B^l \mathbf{c}^l$ . The inverse computation  $\mathbf{c}^l = S^l \mathbf{c}^{l-1} + C^l \mathbf{d}^{l-1}$  is a *synthesis step*. Choosing the linear filters  $A^l$ ,  $B^l$ ,  $S^l$  and  $C^l$  is discussed later. We

\*LSIIT, Université Louis Pasteur, Strasbourg, France,  
e-mail:Basile.Sauvage@dpt-info.i-strasbg.fr,  
https://dpt-info.u-strasbg.fr/sauvage

†Laboratoire Jean Kuntzmann, BP. 53, F-38041 Grenoble (France),  
e-mail:Stefanie.Hahmann@imag.fr,  
http://www-lmc.imag.fr/lmc-mga/Stefanie.Hahmann

‡Laboratoire Jean Kuntzmann, INRIA Rhône-Alpes, 655 Avenue de l'Europe, F-38330 Montbonnot (France),  
e-mail: Georges-Pierre.Bonneau@imag.fr,  
http://www-evasion.imag.fr/Membres/Georges-Pierre.Bonneau

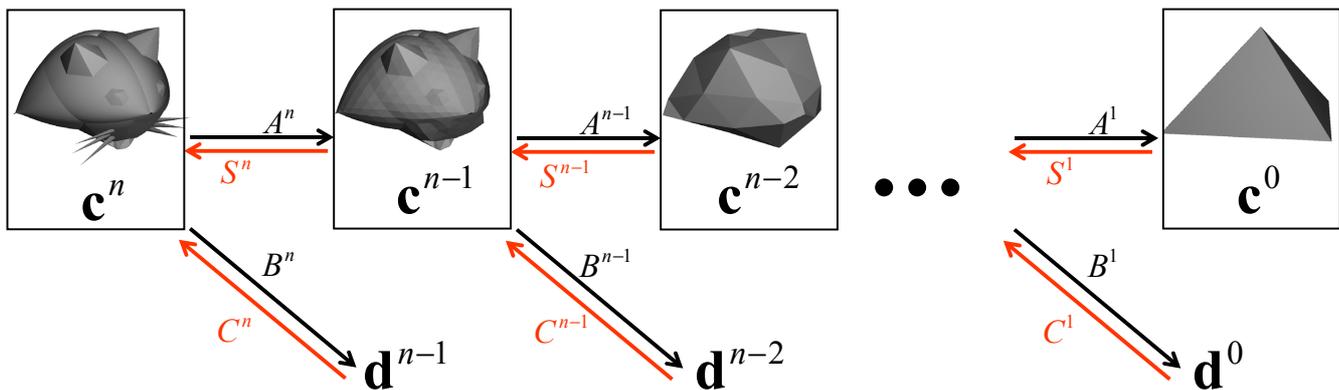


Figure 1: Filterbank. From left to right: analysis process (filters  $A$  and  $B$ ). From right to left: synthesis process (filters  $S$  and  $C$ ).

implemented the MR mesh in a quadtree of faces similar to [4].

Among the numerous advantages of the MR representation, we are mainly interested in the editing abilities. By stopping the analysis process at an arbitrary intermediate level  $0 \leq e \leq n$ , one represents the shape by a coarse mesh  $\mathbf{c}^e$  ( $\mathcal{M}^e$ ) and sequences of details  $(\mathbf{d}^e, \dots, \mathbf{d}^{n-1})$ . Then one can edit the coarse mesh while keeping the details constant. It results in a deformation at the scale  $e$  of the reconstructed mesh while preserving automatically the finer features of the shape.

#### Multiresolution filters.

Since the analysis filters ( $A$  and  $B$ ) and the synthesis filters ( $S$  and  $C$ ) are linear, it is convenient to use matrix notation. In practice however, it is better to use local masks that are matched on the vertices. In particular, the subdivision filter  $S$  corresponds to some subdivision scheme. Thus the corresponding MR scheme is told “to be based on the subdivision scheme”. Several MR schemes for meshes have been proposed in the past. Based on the linear subdivision, the MR schemes proposed by Lounsbery *et al.* [17] tend to lack smoothness for our application. Based on the butterfly subdivision, the main issue is the smoothness around irregular vertices. Then we adapted the lifted wavelets of [21] to the modified Butterfly subdivision [26]. Based on the Loop subdivision, the main issue is the stability of the analysis, and in particular around vertices of low valence. Among several MR schemes [13, 15, 2] the stabilized scheme proposed by Li *et al.* [15] turned out to be the most robust one.

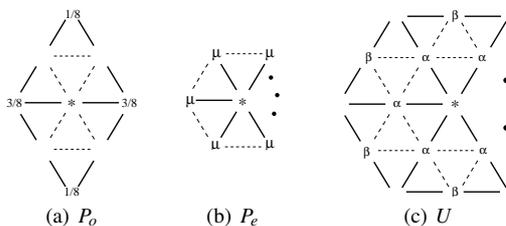


Figure 2: Multiresolution masks. Solid edges stand for the level  $l - 1$ , and dashed edges for the level  $l$ .

#### Lifting scheme.

Every MR scheme previously discussed can be described with the lifting scheme paradigm [23]. It is both a flexible representation

and an efficient algorithm for the filterbank. First, the filters are represented by an accumulation of several masks. Then the properties of the scheme (such as smoothness, orthogonality, or vanishing moments) are conveniently adjusted by additional masks. Second, these masks have very local support. Thus the implementation is easy. Third, it allows fully in-place application of the filters, *i.e.* no extra memory is needed. Eventually it is convenient for the transposed scheme that we develop in Section 3.

Figure 3 represents the stabilized scheme of [15] in the form of a lifting scheme [24]. The property of “in-place computation” is illustrated by the upper and lower branches in Figure 3: the vertices are partitioned into two subsets (resp. called “even” and “odd”), and the masks always modify the vertices of one subset by using the vertices of the other subset only.

One synthesis step  $\mathbf{c}^l = S^l \mathbf{c}^{l-1} + C^l \mathbf{d}^{l-1}$  is represented on the right part of Figure 3. It is processed with the masks of Figure 2. It consists in five phases:

- The “update” filter  $U$ , illustrated in Figure 2(c), is applied. The values of  $\alpha$  and  $\beta$  are given in [15], up to the sign, and up to a normalization by  $1/v_*$ , where  $v_*$  is the valence of the vertex  $*$ .
- The filter  $P_o$  (called “odd predictor”, Figure 2(a)) is applied.
- The normalization step  $N$  multiplies every  $\mathbf{c}_i^{l-1}$  by  $\frac{8}{5}(\frac{3}{8} + \frac{1}{4} \cos(\frac{2\pi}{v_i}))^2$ , where  $v_i$  is the valence of the vertex  $i$ .
- The filter  $P_e$  (called “even predictor”, Figure 2(b)) is applied, with  $\mu = \frac{1}{v_*} - \frac{8}{5v_*}(\frac{3}{8} + \frac{1}{4} \cos(\frac{2\pi}{v_*}))^2$ .
- The merging phase does nothing but considering that the current level is  $l$ .

The phases  $P_o$ ,  $N$  and  $P_e$  define the simplest MR scheme such that the subdivision  $S$  is the Loop subdivision. The adjusting phase  $U$  improves the stability of the scheme by an approximated bi-orthogonalization. Figure 3 also clearly shows the reciprocity of analysis and synthesis: the phases are applied in reverse order and the operators are swapped.

### 3 Volume computation

In the previous section we set the multiresolution framework for the meshes. Constraining the volume of such meshes requires an efficient computation of the volume with respect to the MR representation. It is the purpose of this section, and one of the contributions of the present paper. We derive MR formulas for the volume

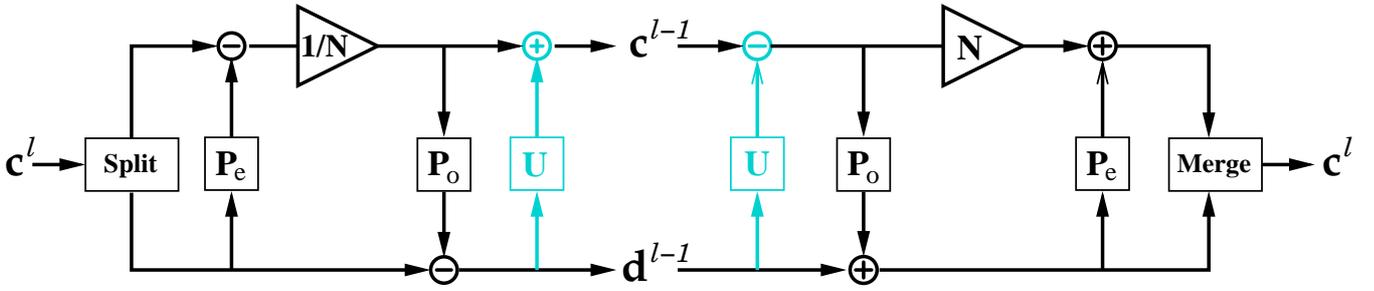


Figure 3: Lifting scheme based on the Loop subdivision scheme. Left: analysis  $c^l \rightarrow (c^{l-1}, d^{l-1})$  represents the filters  $A^l$  and  $B^l$ . Right: synthesis  $(c^{l-1}, d^{l-1}) \rightarrow c^l$  represents the filters  $S^l$  and  $C^l$ .

$\Theta$  enclosed by the mesh while represented at any level of decomposition. We therefore will be able to compute the volume of the fine mesh as a function of the coarse coefficients and details at any level  $e$ . Our approach is an extension to 3D of the area computation proposed in [11]. The mesh is assumed to be closed, manifold, and oriented. Then the signed volume is positive if the normals point outwards.

### 3.1 Recursive formulas

In the following we note  $(x_i^l, y_i^l, z_i^l)$  the three coordinates related to the vertex  $i$  of  $\mathcal{M}^n$  at some level  $l$ . For a fixed  $i$ ,  $1 \leq i \leq N$ , these coordinates exist whatever the level  $l$ , but they may have different meanings. Let  $l_i$  be the depth of  $i$ , i.e. the level such that  $i$  is a vertex of  $\mathcal{M}^{l_i}$  but not  $\mathcal{M}^{l_i-1}$ . If  $l_i \leq l$  then they are the coordinates of  $c_i^l$ , otherwise they are the coordinates of  $d_i^l$ . We also use the vector notations  $x^l = (x_i^l)_i \in \mathbb{R}^N$ ,  $y^l = (y_i^l)_i$ , and  $z^l = (z_i^l)_i$ .

For any given level  $e$  when editing the shape, we want to compute the volume with respect to the MR decomposition at that level. Then the purpose is to write the volume  $\Theta$  as a function of the coordinates  $(x^e, y^e, z^e)$ .

When the mesh is expressed at the finest level  $n$ , the volume is given by the alternating trilinear form

$$\Theta = \sum_{i,j,k} \theta_{ijk}^n x_i^n y_j^n z_k^n, \quad (1)$$

where the sum runs over all the triplets of vertices in  $\mathcal{M}^n$ , and  $\theta_{ijk}^n = 1/6$  if  $ijk$  is a triangle of  $\mathcal{M}^n$ ;  $\theta_{ijk}^n = -1/6$  if  $ikj$  is a triangle of  $\mathcal{M}^n$ ;  $\theta_{ijk}^n = 0$  otherwise. This can be derived from the divergence theorem [9], or, for triangular meshes, by summing (over all the triangles) the (signed) volume of the tetrahedra composed by a triangle of  $\mathcal{M}^n$  and a common fourth vertex (e.g. the origin).

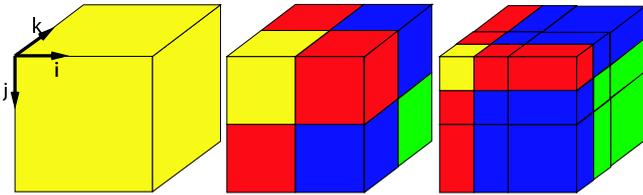


Figure 4: Recursive processing of the volume tensors. From left to right:  $\theta^n$ ,  $\theta^{n-1}$  and  $\theta^{n-2}$ .

Since the filters apply only linear combinations on the coordinates, the same volume is also a trilinear form with respect to the coordinates in the multiresolution basis at level  $e$ :

$$\Theta = \sum_{i,j,k} \theta_{ijk}^e x_i^e y_j^e z_k^e, \quad (2)$$

i.e. the volume can be expressed, for any arbitrary intermediate level  $e$ , with respect to  $x^e$ ,  $y^e$ , and  $z^e$  (vectors containing the coordinates of coarse and detail coefficients, i.e.  $c^e$  and  $d^e, \dots, d^{n-1}$ ). The values  $\theta_{ijk}^e$  are left to be computed.

The sets  $\theta^l = (\theta_{ijk}^l)_{1 \leq i,j,k \leq N}$  are rank 3 tensors, and can be thought of as 3-dimensional arrays (see Figure 4). Directly computing the coefficients of  $\theta^e$  that are used in equation (2) is a hard task. Since the values for  $\theta^n$  are known, we propose an iterative processing  $\theta^n \rightarrow \theta^{n-1} \rightarrow \dots \rightarrow \theta^e$  (see Figure 4). To accomplish this purpose, we derive a recursive formula for  $\theta^{l-1}$  from  $\theta^l$ . Let us start by looking into the first step  $l = n$ :  $\theta^n$  being known, we seek for  $\theta^{n-1}$  satisfying

$$\sum_{i,j,k} \theta_{ijk}^n x_i^n y_j^n z_k^n = \sum_{i,j,k} \theta_{ijk}^{n-1} x_i^{n-1} y_j^{n-1} z_k^{n-1},$$

which is obtained by equating (1) and (2). From the filterbank we know that  $x^{n-1} = \begin{bmatrix} A^n \\ B^n \end{bmatrix} x^n$ . The same relation holds for  $y$  and  $z$ . We also have the inversion relation on the square matrices  $\begin{bmatrix} A^n \\ B^n \end{bmatrix} = [S^n \ C^n]^{-1}$ , since the MR scheme we use ensures exact reconstruction. Thus  $\theta^{n-1}$  is derived from  $\theta^n$  by multiplying, for every axis ( $x$ ,  $y$  and  $z$ ), every row by  $[S^n \ C^n]^T$ . This first step is shown Figure 4 (from left to middle). With a similar reasoning one proves that the computation of  $\theta^{l-1}$  from  $\theta^l$ , for any level  $l$ , ensues from the application of the transposed filters  $(S^l)^T$  and  $(C^l)^T$  in a tensor product manner:

- Following the 3 axes on the yellow (light) block,
- following the 2 axes on the red (mid grey) blocks,
- following the 1 axis on the blue (dark) blocks,
- and green blocks (on the right, down and below) are invariant.

### 3.2 Computational issues

In order to compute the volume in the MR basis through formula (2), we derived, in the previous section, recursive formulas for computing the volume tensor  $\theta^e$  from  $\theta^n$ . Two algorithmic challenges still hinder the implementation of this recursive process. First, the tensors  $\theta^l$  have size  $N^3$ , where  $N$  is the number of vertices. Second, the filters  $S^l$  and  $C^l$  need to be transposed, but they are encoded as masks, not as matrices.

Regarding their size, the  $\theta^l$  can not be stored in full-format. We chose a sparse storage that meets two requirements: i) it must be dynamic because new coefficients are inserted when the transposed filters are applied, and ii) the traversal of the structure must be efficient in any direction because of the computation of the sums in

formula (2). Thus we use an orthogonal triple chaining of the non-zero elements which generalizes the double chaining for usual matrices (i.e. rank 2 tensors).

Moreover  $\theta^l$  has several symmetries (namely, it is an antisymmetric tensor), thus the number of stored elements can be reduced at least by a factor of 3. Eventually, restricting the volume preservation only to a subset  $\mathcal{F}$  of vertices (see section 4), further elements can be removed. For typical instance, if  $\mathcal{F}$  contains only vertices of  $\mathcal{M}^e$  and no detail coefficients, all the green blocks in Figure 4 can be removed.

We still know no theoretical bound for the sparsity of the tensors. However experimental results are significant. The horse animation (see Figure 8) involves  $5E6$  non-zero coefficients out of  $N^3 = 1.2E9$ , but only  $4.5E6$  need to be stored. The bouncing ball animation (see Figure 6) involves  $42E6$  non-zero coefficients out of  $N^3 = 265E6$ , but only  $4E6$  need to be stored.

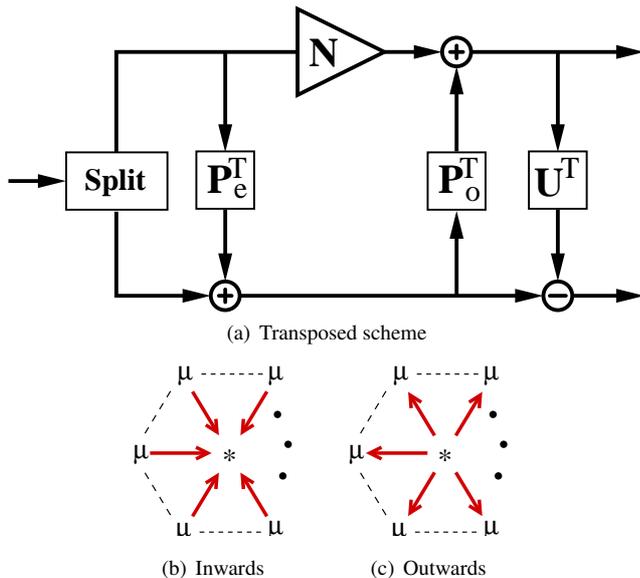


Figure 5: Transposed lifting scheme. (b) Inwards application of the filter  $P_e$  for the usual Lifting scheme. (c) Outwards application for the transposed scheme (a).

The second challenge consists in transposing the synthesis filters  $S^l$  and  $C^l$ . Since we are not using the matrices  $S^l$ ,  $C^l$  for synthesis but the filters of the lifting scheme, we need to transpose the right part of Figure 3. The “transposed lifting scheme” is illustrated in Figure 5(a). It is obtained from the direct scheme by i) reversing the order the masks are applied, ii) keeping the operators, iii) transposing the masks. Here “transposing” the masks should be understood as: all the masks are applied outwards instead of inwards, see Figure 5(b) and 5(c). The masks  $P_e$  and  $U$  are still instantiated for all the vertices of  $\mathcal{M}^{l-1}$ , while  $P_o$  is still instantiated for all the new vertices of  $\mathcal{M}^l \setminus \mathcal{M}^{l-1}$ . For instance, let apply  $P_e$  instantiated with the central vertex  $* = v$  of  $\mathcal{M}^{l-1}$ . The *inwards* application (i.e. during the reconstruction  $(\mathbf{c}^{l-1}, \mathbf{d}^{l-1}) \rightarrow \mathbf{c}^l$ , see Figure 3 right) means

$$\mathbf{c}_v^{l-1} = \mathbf{c}_v^{l-1} + \mu \sum_{\omega} \mathbf{d}_{\omega}^{l-1},$$

where the sum runs over the neighbors of  $v$ . In contrast the *outwards* application on the row  $\theta_{ij}^l$ , for some fixed  $i$  and  $j$ , means (see Figure 5)

$$\theta_{ij\omega}^l = \theta_{ij\omega}^l + \mu \theta_{ijv}^l \quad \forall \omega \text{ neighbors of } v.$$

Now,  $\theta^{l-1}$  can be computed from  $\theta^l$  by applying the transposed lifting scheme to all rows of the tensor  $\theta^l$  in  $x$ ,  $y$  and  $z$  direction successively. Note that once one row has been transformed it is replaced in the tensor before processing the next row.

Then the algorithm for the recursive pre-processing of  $\theta^e$  is completed. It is expensive but it is applied only once, and  $\theta^e$  is stored before the deformation of the surface. For all the examples shown here, the pre-processing lasts a couple of minutes.

## 4 Volume preserving deformation

In the previous section we established formulas and an efficient algorithm for computing the volume enclosed by a MR mesh at any level of resolution. In the present section we use it for preserving the volume during a deformation process. We assume that the level  $e$  is fixed (chosen by the user), that the surface is decomposed at this level, and that  $\theta^e$  has been pre-computed. Since the deformation process is fully defined with respect to the MR representation, the superscript  $e$  is omitted from now on.

### 4.1 Problem statement

We do not make any assumption about the way the deformation is driven (e.g. drag & drop of the vertices in a modeler, or skeleton driven deformation, FFD, etc.). Such a genericity is achieved by inputting an arbitrary target shape and outputting a correction of this shape such that a prescribed volume is met. In return we assume that the deformation is small (later needed for a linear approximation of the volume variation). This assumption is reasonable when modeling a continuous deformation of the mesh, but large deformations must be divided into small steps.

Thus we state the problem (for each small step) as follows:

- $\Theta_{ref}$  is the reference volume that will be preserved. Note that  $\Theta_{ref}$  is usually the volume of the input mesh, but can also be set to any arbitrary value.
- A target shape  $\tau = (x_i^{\tau}, y_i^{\tau}, z_i^{\tau})_i$  is given by some deformation process. Its volume  $\Theta_{\tau}$  is a priori not equal to  $\Theta_{ref}$ .
- A set  $\mathcal{F} \subset \{1, \dots, N\}$  of coefficients is selected. Only the coefficients in  $\mathcal{F}$  are free to change in order to enforce the volume preservation. In many situations the MR representation is exploited by filling  $\mathcal{F}$  only with vertices at the current level of resolution, while all the fine details are kept fixed during the deformation. Moreover setting  $\mathcal{F}$  allows to localize the deformation.
- We seek for a minimal correction  $\delta$  bounded to  $\mathcal{F}$ , such that the final mesh  $\tau + \delta$  encloses a volume  $\Theta_{\tau+\delta} = \Theta_{ref}$ . This correction is defined by displacements  $\delta_i \in \mathbb{R}^3$ :

$$\begin{pmatrix} x_i^{\tau} \\ y_i^{\tau} \\ z_i^{\tau} \end{pmatrix} \rightarrow \begin{pmatrix} x_i^{\tau} \\ y_i^{\tau} \\ z_i^{\tau} \end{pmatrix} + \delta_i \quad \forall i \in \mathcal{F}.$$

### 4.2 Constrained minimization

For the purpose of defining  $\delta$ , we have two informations at our disposal: i) it is suitable for  $\delta$  to be as small as possible, in order to keep the final mesh as close as possible to the target; ii) the final volume is required to equal  $\Theta_{ref}$ . Therefore we propose to define  $\delta$  by the constrained minimization

$$\min \sum_{i \in \mathcal{F}} \|\delta_i\|^2 \quad \text{subject to } \Theta_{\tau+\delta} = \Theta_{ref}.$$

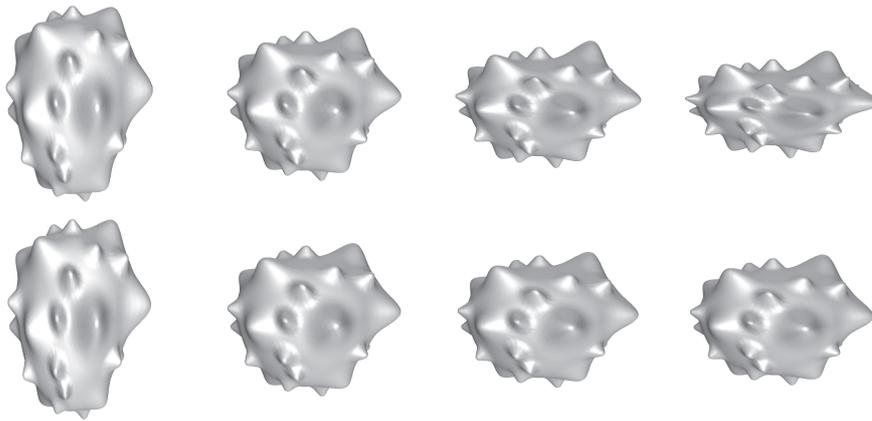


Figure 6: Several deformations of a spiky ball are extracted from an animation of a bouncing ball. Upper row: the volume changes. Lower row: preserving the volume improves the realism. The control mesh is an icosahedron. Note how the details are preserved thanks to the MR representation.

By plugging the coordinates of  $\tau + \delta$  into equation (2), and keeping only the first order terms (with respect to the coordinates of the  $\delta_i$ ), one gets the following linear approximation of the volume constraint:

$$\Theta_{ref} - \Theta_\tau \approx \sum_{i \in \mathcal{F}} \Theta_\tau(i) \cdot \delta_i \quad (3)$$

where  $\Theta_\tau(i) \in \mathbb{R}^3$  is defined for any  $1 \leq i \leq N$  by the partial sums in formula (2) :

$$\Theta_\tau(i) = \left( \sum_{j,k=1}^N \theta_{ijk} y_j^\tau z_k^\tau, \sum_{j,k=1}^N \theta_{jik} x_j^\tau z_k^\tau, \sum_{j,k=1}^N \theta_{jki} x_j^\tau y_k^\tau \right)^T.$$

Replacing the volume constraint by its linear approximation (3) leads to minimizing a quadratic form with respect to one linear constraint. The solution is known to be a saddle point of the Lagrangian

$$g((\delta_i)_{i \in \mathcal{F}}, \lambda) = \sum_{i \in \mathcal{F}} \|\delta_i\|^2 + \lambda \left( \sum_{i \in \mathcal{F}} \Theta_\tau(i) \cdot \delta_i - (\Theta_{ref} - \Theta_\tau) \right).$$

Thus it satisfies the linear system  $\nabla g = 0$  :

$$\begin{aligned} 2\delta_i + \lambda \Theta_\tau(i) &= 0 & \forall i \in \mathcal{F}, \\ \sum_{i \in \mathcal{F}} \Theta_\tau(i) \cdot \delta_i &= \Theta_{ref} - \Theta_\tau. \end{aligned} \quad (4)$$

By inserting (4) into (5), and then inserting  $\lambda$  back into (4), one prove that the solution has the closed form

$$\lambda = \frac{-2(\Theta_{ref} - \Theta_\tau)}{\sum_{j \in \mathcal{F}} \|\Theta_\tau(j)\|^2}$$

and for all  $i \in \mathcal{F}$

$$\delta_i = \frac{\Theta_{ref} - \Theta_\tau}{\sum_{j \in \mathcal{F}} \|\Theta_\tau(j)\|^2} \Theta_\tau(i). \quad (6)$$

The formula (6) is the key of a real-time volume preserving deformation process. It is a closed form that solves the constrained minimization. It thus concentrates all the computational effort at each time step. Computing  $\Theta_\tau$  is the most expensive: it costs in proportion to the fill-in of  $\theta^\ell$ . The cost can be further reduced because computing the volume change  $\Theta_{ref} - \Theta_\tau$  is cheaper than computing the volume itself. Indeed, the only useful terms in formula (2) are

those involving a moving vertex. Thus the efficiency increases dramatically in a frequent situation, namely when the coarse vertices are moved while the details are kept fixed.

The proposed process enforces the approximation (3) of the volume constraint. Thus an error is made at each step but they don't cumulate since  $\theta_{ref}$  is fixed. Then the error is controlled before proceeding to the next step: the correction (6) is iterated until a specified relative threshold is met. Experiments shown that an iteration was seldom necessary for reasonably small steps. We used  $10^{-5}$  as a threshold in our examples. Even with this small relative error, the volume preserving deformation is still performed in real-time.

## 5 Results

We present here several examples that show the effectiveness of our volume preserving algorithm. It is combined with various deformation techniques, which proves the versatility of our method. All examples undergo first a MR analysis in order to obtain a coarse control mesh. The scheme based on the modified Butterfly subdivision is used for the horse and the bunny examples. The scheme based on the Loop subdivision scheme is used for the bouncing ball example. We then apply the deformations to the control mesh while preserving the volume of the fine input mesh. Once the volume coefficients  $\theta^\ell$  (see Section 3.1) are pre-computed and stored, all deformations and animations can be performed in real-time.

In Figure 6 are shown several deformations of a spiky ball. Thanks to the MR representation ( $N = 642$ ,  $n = 3$  and  $e = 0$ ) the details are preserved while the global shape changes. The volume is free in the upper row, and preserved in the lower row. If modeling a soft but non-stretch material, the volume preservation obviously improves the realism of the deformations. Those positions are extracted from an animation of a bouncing ball<sup>1</sup>. The coarse control mesh (an icosahedron) was animated thanks to physical 1D equations.

The Stanford bunny is deformed in Figure 7. From the initial position (upper right), the ears are deformed by direct manipulation of a few vertices of the control mesh (upper left) in a 3D editing tool. Several resulting positions are shown in the lower row. Remark how the proportions are preserved thanks to the volume preservation, and how the features are preserved thanks to the multiresolution ( $N = 2002$ ,  $n = 2$  and  $e = 0$ ).

In Figure 8 a horse model is animated ( $N = 1762$ ,  $n = 2$  and

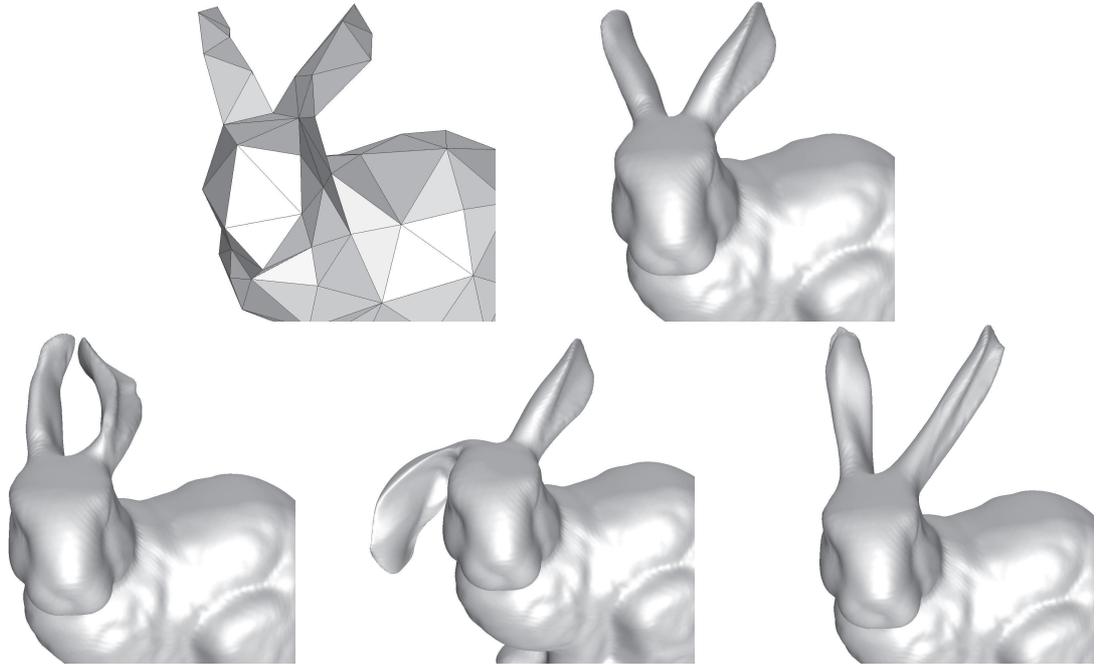


Figure 7: The ears of the Stanford bunny are moved while preserving their volume. The initial mesh (upper right) is modified by directly manipulating the vertices of its control mesh (upper left). The proportions in the resulting positions (lower row) are preserved thanks to the volume constraint. The MR representation automatically preserves the fine features.

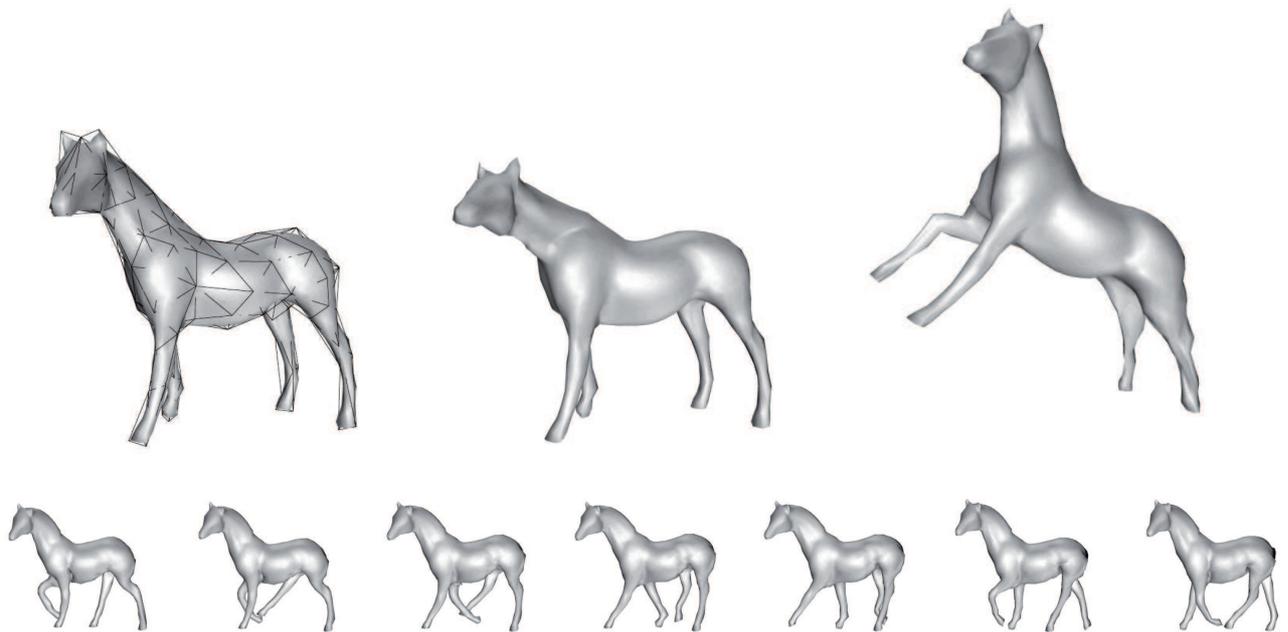


Figure 8: The horse is animated by skinning, which is combined with the MR mesh and our volume preserving method. The MR mesh is deformed through its control mesh (wireframe upper left) according to a skeleton. Upper row: initial mesh (left) and various positions. The lower row show several frames extracted from a walking animation.

$e = 0$ ). Here the volume preservation was combined with a skinning model, *i.e.* the control mesh (wireframe in the upper left) was animated thanks to a skeleton. The lower row shows several positions that are extracted from a walking animation<sup>1</sup>.

## 6 Conclusion and future works

A method for computing the volume enclosed by a multiresolution mesh has been presented. It is expressed as a trilinear form whatever the level of decomposition. The volume computation is made efficient thanks to a recursive pre-processing using a sparse data structure. Therefore we proposed a new method for transposing linear filters when represented by local masks through the lifting scheme model. Then we derived an algorithm for preserving the volume while deforming the mesh through its MR decomposition. It is based on a linearization of the volume constraint that is used in a quadratic minimization process. A closed form of the solution makes it efficient. A simple design makes it suitable for a broad range of handling tools. In particular, some examples show convincing couplings with a direct editing tool and with a skinning deformation model. In the following we discuss a few existing extensions and sketch some ideas for future work.

**Meshes with boundaries.** All the results presented in this paper also work for meshes with boundaries, as long as they are manifold and that masks for the boundaries are available. A particular attention should however be paid to keep the boundaries fixed during the deformation. Otherwise preserving the volume is absurd.

**Smooth surfaces.** For the present application to computer graphics we focused on meshes only. However the connection to wavelet theory reminds that the MR decompositions also represents a continuous surface with respect to the wavelet basis. This surface is the limit of the subdivision process applied to the fine mesh  $\mathbf{c}^n(\mathcal{M}^n)$ . All the results hold for computing and preserving the volume of the limit surface instead of the fine mesh. Only the initial coefficients in  $\theta^n$  should be modified by computing integrals over the surface. A similar approach is detailed in [11] about the area of planar curves.

**Data structures.** For the moment, the size of the meshes is limited to a few thousand vertices because of the size of the tensors. Though  $\theta^n$  has  $O(N)$  non-zero entries only,  $\theta^e$  is more filled. Moreover the chained data structure proposed in section 3 requires at least 26 bytes per entry. It could therefore be worthwhile to use it during the recursive process only, and move to a static but more compact representation afterward.

**Localized volume preserving.** Unlike previous works [3, 16] our method is not based on local volume elements. Thus, when processing global deformations of the mesh, the volume correction might have unnatural behavior. We investigate the use of a partition of the mesh in order to preserve separately the volume of each part.

**Global versus local frame.** On one hand, encoding the details in a global frame is known not to be rotation invariant but has the advantage of making the analysis and synthesis filters linear. This is a requirement of the present volume preservation method. On the other hand, encoding details in local frames preserves the orientation of the details, and is therefore more appropriate for large scale deformations.

Using local frames is a challenging future work since the filterbank is not linear any more.

<sup>1</sup>See the accompanying video.

**Higher order moments.** The extension to higher order moments [9] is very promising for future research. For instance the center of mass and the inertia tensor could be computed in the MR basis. Indeed, the machinery used here only assumes that the computed quantity is multi-linear. Up to the memory requirements, the extension is then straightforward.

## Acknowledgements

The work was partially supported by the AIM@SHAPE Network of Excellence (FP6 IST NoE 506766) and the IMAG project MéGA. The author will also thank Thomas Carcaud for his help. The horse is courtesy of Cyberware. The bunny is courtesy of Stanford University. The semi-regular versions of these meshes have been provided by the Caltech Multi-Res Modeling Group.

## References

- [1] AUBERT, F., AND BECHMANN, D. Volume-preserving space deformation. *Computers and Graphics* 21, 5 (Sept.–Oct. 1997), 625–639.
- [2] BERTRAM, M. Biorthogonal loop-subdivision wavelets. *Computing* 72, 1-2 (2004), 29–39.
- [3] BOTSCH, M., AND KOBBELT, L. Multiresolution surface representation based on displacement volumes. *Computer Graphics Forum* 22, 3 (2003), 483–491. (Proceedings Eurographics '03).
- [4] CERTAIN, A., POPOVIĆ, J., DE ROSE, T., DUCHAMP, T., SALESIN, D., AND STUETZLE, W. Interactive multiresolution surface viewing. *SIGGRAPH Computer Graphics* (1996), 91–98.
- [5] ELBER, G. Multiresolution curve editing with linear constraints. In *6th ACM/IEEE Symposium on Solid Modeling and Applications* (June 2001), Ann Arbor, Michigan, pp. 109–119.
- [6] FINKELSTEIN, A., AND SALESIN, D. H. Multiresolution curves. *SIGGRAPH Computer Graphics* (1994), 261–268.
- [7] FOWLER, B. Geometric manipulation of tensor product surfaces. In *Proceedings of the 1992 symposium on Interactive 3D graphics* (1992), ACM Press, pp. 101–108.
- [8] FOWLER, B., AND BARTELS, R. Constraint-based curve manipulation. *IEEE Computer Graphics and Applications* 13, 5 (1993), 43–49.
- [9] GONZALEZ-OCHOA, C., MCCAMMON, S., AND PETERS, J. Computing moments of objects enclosed by piecewise polynomial surfaces. *Transactions on Graphics* 17, 3 (1998), 143–157.
- [10] GUSKOV, I., VIDIMCE, K., SWELDENS, W., AND SCHRÖDER, P. Normal meshes. *SIGGRAPH Computer graphics* (2000), 95–102.
- [11] HAHMANN, S., SAUVAGE, B., AND BONNEAU, G.-P. Area preserving deformation of multiresolution curves. *Computer-Aided Geometric Design (CAGD)* 22, 4 (2005), 349–367.
- [12] HIROTA, G., MAHESHWARI, R., AND LIN, M. C. Fast volume-preserving free form deformation using multi-level optimization. In *SMA '99: Proceedings of the fifth ACM symposium on Solid modeling and applications* (New York, NY, USA, 1999), ACM Press, pp. 234–245.

- [13] KHODAKOVSKY, A., SCHRÖDER, P., AND SWELDENS, W. Progressive geometry compression. *SIGGRAPH Computer Graphics* (2000), 271–278.
- [14] KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H.-P. Interactive multi-resolution modeling on arbitrary meshes. *SIGGRAPH Computer graphics* (1998), 105–114.
- [15] LI, D., QIN, K., AND SUN, H. Unlifted loop subdivision wavelets. In *Pacific Graphics Conference on Computer Graphics and Applications* (2004), pp. 25–33.
- [16] LIPMAN, Y., COHEN-OR, D., GAL, R., AND LEVIN, D. Volume and shape preservation via moving frame manipulation. *ACM Transactions on Graphics to appear* (2007).
- [17] LOUNSBERY, M., DEROSE, T., AND WARREN, J. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transaction on Graphics* 16, 1 (1997), 34–73.
- [18] MALLAT, S. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (1989), 674–693.
- [19] PAYAN, F., AND ANTONINI, M. Mean square error approximation for wavelet-based semiregular mesh compression. *Transactions on Visualization and Computer Graphics (TVCG)* 12, 4 (2006).
- [20] RAPPOPORT, A., SHEFFER, A., AND BERCOVIER, M. Volume-preserving free-form solids. In *IEEE Transactions on Visualization and Computer Graphics* (1996), vol. 2(1), pp. 19–27.
- [21] SCHRÖDER, P., AND SWELDENS, W. Spherical wavelets: efficiently representing functions on the sphere. *SIGGRAPH Computer Graphics* (1995), 161–172.
- [22] STOLLNITZ, E. J., DEROSE, T., AND SALESIN, D. H. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1996.
- [23] SWELDENS, W. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.* 29, 2 (1997), 511–546.
- [24] SWELDENS, W., AND SCHRÖDER, P. Building your own wavelets at home. In *Wavelets in Computer Graphics, SIGGRAPH Course notes*. ACM Press, 1996, pp. 15–87.
- [25] VON FUNCK, W., THEISEL, H., AND SEIDEL, H.-P. Vector field based shape deformations. *SIGGRAPH Computer Graphics* (2006), 1118–1125.
- [26] ZORIN, D., SCHRDER, P., AND SWELDENS, W. Interpolating subdivision for meshes with arbitrary topology. *SIGGRAPH Computer Graphics* 30 (1996), 189–192.
- [27] ZORIN, D., SCHRDER, P., AND SWELDENS, W. Interactive multiresolution mesh editing. *SIGGRAPH Computer Graphics* (1997), 259–268.