FROM TRIANGULATION TO SIMPLEX MESH, AND VICE-VERSA
a Simple and Efficient Conversion

Francisco J. GALDAMES[1,3] and Fabrice JAILLET[2,3]
[1]*Department of Electrical Engineering, Universidad de Chile, Av. Tupper 2007, Santiago, Chile*
[2]*Université de Lyon, IUT Lyon 1, Computer Science Department, F-01000, France*
[3]*Université de Lyon, CNRS, Université Lyon 1, LIRIS, SAARA team, UMR5205, F-69622, France*
{francisco.galdames, fabrice.jaillet}@liris.cnrs.fr

Keywords: Simplex Mesh : Triangle Mesh : Optimized Surface Interpolation : Surface Mesh Conversion.

Abstract: In the field of 3D images, relevant information can be difficult to interpret without further computer-aided processing. Generally, and this is particularly true in medical imaging, a segmentation process is run and coupled with a visualization of the delineated structures. Amongst all techniques based on deformable models, *simplex meshes*, in particular, present good propensities to handle a large variety of shape alterations altogether with a fine resolution and stability. However, they may not be well suited to cope satisfyingly with other related tasks, such as rendering, mechanical simulation or reconstruction from iso-surfaces. As a consequence, *triangle meshes* are often preferred. Thus, we propose an accurate method to shift from a model to another, and conversely. For this, we are taking advantage of the fact that they are topologically duals, turning it into a natural swap between these two models. Unfortunately, they are not geometrically equivalents, leading to loss of information and to geometry deterioration when performing the conversion. Therefore, optimal positions of the vertices in the dual mesh have to be found while avoiding shape degradation. An accurate and effective transformation technique is described in this paper, where we present a direct method to perform an appropriate interpolation of a simplex mesh to obtain its dual, and/or vice-versa. This original method is based on the distance minimization between the local tangent planes of the mesh and vertices of each face. Finally, probing resulting mesh conversions in both directions are commented.

1 INTRODUCTION

Deformable model techniques are widely used in image segmentation tasks. Among these models, it is indubitable that simplex meshes are valuable candidates (Delingette, 1999), for their favorable characteristics in this type of modeling, as its easy control and convenient way to model internal forces. With this type of meshes, as with triangulations, any topology can be described. Furthermore, simplex meshes and triangulations are topologically duals, and this allows us to naturally obtain a simplex mesh by applying a dual operation to the triangulation, and vice-versa. On the other hand, if very efficient algorithms exist to generate triangulations from a given geometry (Treece et al., 1999; Lorensen and Cline, 1987), this can be more arduous using simplex meshes. So, in some cases it could be appropriate to generate a preliminary triangulation (from an isosurface, for example), and next to transform this latter into a simplex mesh, with the intention of controlling the deformation of the model (Xu et al., 1999).

Moreover, there are some applications for which simplex meshes are not suitable, and thus triangulated meshes are more adapted. For example, there are different meshing methods to generate inner tetrahedral or hexahedral elements (Owen, 1998; Si, 2010), but in general they need as input a surface representation that clearly defines the geometry, with planar faces and without self-intersections between elements. Piecewise linear complex (PLC) (Barber et al., 1996) and triangulation are eligible. However, simplex meshes do not meet these requirements since faces are not necessarily planar. One possibility would be to transform the simplex mesh into a triangulation before creating the volumetric mesh. Rendering and calculation of area, may be other tasks where it is preferable to handle triangulations instead of simplex meshes.

Triangulations and simplex meshes are topologically duals, but not geometrically equivalents (Delingette, 1999). No homeomorphism can be built between the set of coordinates of a triangulation and the one of its dual simplex mesh. Therefore, there is often loss of information and geometry deterioration whenever a transformation between these meshes takes place. Numerous algorithms exist to perform what is called *remeshing* (Alliez et al., 2008). Nevertheless, they aim in general to build or adapt triangle meshes on cloud points, but they are not suited to deal with the particular case of simplex meshes, and even less to convert triangulations into simplex meshes.

Currently, the most common way to perform this conversion is to determine the set of vertices for the final mesh as the gravity center of each face of the initial mesh. This technique is very fast, but unfortunately in this case, mesh smoothing is generally very high; original shape (curvature) and volume is far to be accurately respected. An alternative is to compute the gravity center of each face and next insert this point in the mesh before triangulation. Although this method reduces the alteration in the geometry, the resulting mesh is not dual to the initial simplex mesh, and moreover, the number of vertices will rise considerably. It is also possible to consider only the face vertices, but the resulting mesh will either not be topologically dual. Moreover, the converse process to obtain a simplex mesh from a triangulation is not straightforward. In (de Putter et al., 2006), the authors show the importance of such a transformation, especially in medical applications where simplex meshes are of great use in the creation of the computational mesh based on the segmented geometry. They propose an iterative curvature correction algorithm for the dual triangulation of a two-simplex mesh. Their solution provide optimal error distribution between the two dual surfaces while preserving the geometry of the mesh, but at the price of an iterative global minimization over the whole meshes.

For all these reasons, it is essential to have an efficient method to perform transformations between these two types of meshes. In this paper, a new technique is presented, achieving reasonable computation cost and minimal loss of geometric information. From a geometric point of view, the problem can be reduced to find an interpolation of the center of each face, and to build the dual mesh accordingly to these points. Subdivision, variational surfaces, traditional splines or implicit surfaces are amongst the most used techniques to find interpolating points in a mesh. As the requirement here is to get a simple and straightforward method, we propose to use a geometric interpolation, based on the distance to the tangent planes of the vertices of each face. A similar measure has been successfully used in (Ronfard and Rossignac, 1996) to compute a local geometric error based on the maximal distance to a set of planes, in order to perform triangular mesh simplifications. An equivalent measure has been employed, using this time a summation to obtain a quadratic error (Garland and Heckbert, 1997; Heckbert and Garland, 1999). In a more recent work, a method for refining triangulations has been developed (Yang, 2005). It is based on face splitting and interpolation using distance minimization over the neighboring triangles planes. Here, it is worth to point out that our global objective is to perform a transformation between meshes, and not to refine them. However, we mainly got inspiration from this last work, but in our case the error measurement is applied to find the vertices of a dual mesh, to permit conversion between simplex meshes and triangulations, and conversely.

The paper is organized as follows. In section 2, we present essential background on simplex meshes, their characteristics and relationship with triangulations. The main part concerning the interpolation method used to find the dual mesh is explained in section 3. Application of this method to swap between meshes is shown in sections 4 and 5, where details can be found for each swap direction. Finally, some results are exhibited in section 6, followed by conclusions in 7.

2 TRIANGULATION VS. SIMPLEX MESH

As stated in the introduction, a simplex mesh can be seen as the topological dual of a triangulation, each vertex of the simplex mesh corresponding to a triangle in the dual triangulation (Fig. 1). However, simplex meshes and triangulations are not geometrically duals. Their geometry is determined by the coordinates of their vertices; nevertheless, the number of vertices is different between a simplex meshe V_S and a triangulation V_T. The Euler's characteristic for a triangulation without holes and its dual simplex mesh states:

$$V_T - \frac{V_S}{2} = 2(1-g), \qquad (1)$$

where g is the genus of the mesh. As the sets of coordinates have different dimensions for a triangulation and its dual simplex mesh, no homeomorphism can be constructed between them.

Simplex meshes are privileged candidates to be used in segmentation methods based on deformable models. Each vertex of a simplex mesh has three neighbors $p_{N1(i)}, p_{N2(i)}, p_{N3(i)}$; between them, a restricted number of entities is defined, the simplex angle and the metric parameters. The simplex angle ρ_i

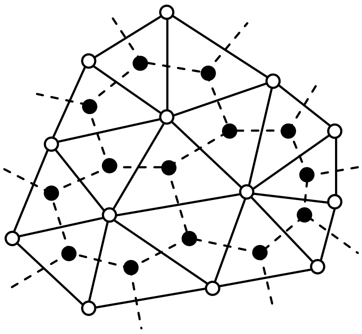


Figure 1: Simplex meshes and triangulations are topological but not geometrical duals. White dots: triangulation vertices ; Black dots: simplex mesh vertices.

is defined at each vertex p_i by means of its neighborhood. The normal vector $\overrightarrow{N_i}$ to the plane defined by the three neighbors, the circle of center C_i and radius r_i defined by these neighbors, and the sphere of center O_i and radius R_i defined by the four vertices can easily be computed. Thus, the simplex angle can be defined as (Fig.2):

$$\begin{aligned} &\rho \in [-\pi, \pi] \\ sin(\rho_i) &= \frac{r_i}{R_i} sign\left(\overrightarrow{p_i p_{N1(i)}} \cdot \overrightarrow{N_i}\right) \\ cos(\rho_i) &= \frac{\|O_i C_i\|}{R_i} sign\left(\overrightarrow{O_i C_i} \cdot \overrightarrow{N_i}\right) \end{aligned} \tag{2}$$

The simplex angle can be considered as a measure of the height of vertex p_i with respect to the plane defined by its neighbors. The metric parameters $\varepsilon_{1i}, \varepsilon_{2i}, \varepsilon_{3i}$ describe the relative position of a vertex according to its neighbors. The position of the projection $p_i^{\perp}$ of vertex p_i on the plane defined by its neighbors (Fig. 2) can be expressed as:

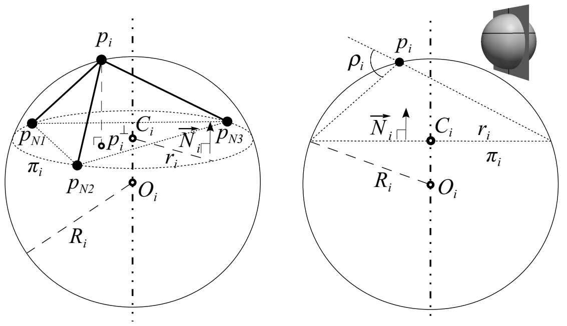


Figure 2: a) Tetrahedron formed by vertex p_i and its 3 neighbors $p_{N1(i)}, p_{N2(i)}, p_{N3(i)}$, with the circle defined by the neighbors, and the sphere containing those four vertices. b) Projection on a plane passing through O_i, C_i and p_i, revealing the simplex angle.

$$\begin{aligned} &p_i^{\perp} = \varepsilon_{1i} p_{N1(i)} + \varepsilon_{2i} p_{N2(i)} + \varepsilon_{3i} p_{N3(i)} \\ &\varepsilon_{1i} + \varepsilon_{2i} + \varepsilon_{3i} = 1 \end{aligned} \tag{3}$$

To perform transformations in any direction between these two types of dual meshes, we have to find an associated vertex q_u of the dual mesh M_2 for each face f_u of the initial mesh M_1. When dealing with triangulations, faces are triangles; and conversely for simplex meshes, faces are polygons whose vertices are generally not coplanar. The resulting mesh M_2 should have a regular shape and preserve the geometry defined by M_1, what is far from being straightforward. For trying to maintain the geometry, we can impose that q_u remains close to the tangent planes π_i of each vertex p_i defining the face f_u. Constraining M_2 to have a regular shape, can be achieved by choosing q_u close to the center of the face f_u, *i.e.* minimize the distance between q_u and all p_i. Therefore, we must minimize the distance between a point q_u and a set of points and planes. Accordingly, the purpose of the present method is to compensate the lack of existing techniques on these aspects.

3 INTERPOLATION BASED ON TANGENT PLANES

The equation of a plane can be denoted as $A \cdot p = 0$, where $A = [a, b, c, d]$ and $p = [x_p, y_p, z_p, 1]^T$ is a point lying on this plane. The coefficients a, b, c are the components of the unit vector $\overrightarrow{N}$ normal to the plane, and $d = -\overrightarrow{N} \cdot p$. For q an arbitrary point in the space, $|A \cdot q|$ is the distance to the plane.

Considering now a set of planes π_i represented by $A_i \cdot p = 0$ $(i = 1, \ldots, L)$, the distance between any point $q = [x, y, z, 1]^T$ to each plane π_i is $|A_i \cdot q|$. On the other hand, let's consider a set of points p_j $(j = 1, \ldots, M)$. If we want to find the point q minimizing its distance to planes π_i and points p_j, the function to be considered follows:

$$D(q) = \sum_{i=1}^{L} \alpha_i |A_i \cdot q|^2 + \sum_{j=1}^{M} \beta_j |q - p_j|^2 \tag{4}$$

where α_i and β_j are weights for the distance to the planes (in order to respect geometry and curvature) and points (controlling shape regularity), respectively. Equation (4) can be rewritten in matrix form as:

$$D(q) = q^T Q q \tag{5}$$

where

$$Q = \sum_{i=1}^{L} \alpha_i A_i^T A_i + \sum_{j=1}^{M} \beta_j Q_j \tag{6}$$

and

$$Q_j = \begin{bmatrix} 1 & 0 & 0 & -x_j \\ 0 & 1 & 0 & -y_j \\ 0 & 0 & 1 & -z_j \\ -x_j & -y_j & -z_j & x_j^2 + y_j^2 + z_j^2 \end{bmatrix} \tag{7}$$

Since Q_j and $A_i^T A_i$ are symmetric matrices, then Q is also symmetric and can be written as:

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \tag{8}$$

To minimize the quadratic form of eq. (5), let's solve the following system of equations:

$$\frac{\partial D(q)}{\partial x} = 0, \; \frac{\partial D(q)}{\partial y} = 0, \; \frac{\partial D(q)}{\partial z} = 0. \tag{9}$$

Taking the partial derivatives of:

$$\begin{aligned} q^T Q q = {} & q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x + q_{22}y^2 \\ & + 2q_{23}yx + 2q_{24}y + q_{33}z^2 + 2q_{34}z + q_{44}, \end{aligned} \tag{10}$$

it can be noticed that the system in eq. (9) can be rewritten in a matrix form as:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{11}$$

Finally, the solution of eq. (11) follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{12} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix}^{-1} \begin{bmatrix} -q_{14} \\ -q_{24} \\ -q_{34} \end{bmatrix} \tag{12}$$

where $q = [x, y, z]^T$.

Weights calculation.

The solution of equation (4) can be understood as an affine combination of the generalized intersection of all planes π_i (first term) and the average of all points p_j (second term). This affine combination is controlled by the weights α_i and β_i. For example, let's consider points p_1, p_2 and planes π_1, π_2 as shown on Figure 3. Planes intersect at point p_α, and the average of the points (for $\beta_i = \beta$) is p_β. The weights α_i should reflect the importance of each plane to the interpolation; and this importance will be estimated in a different way for triangulations or simplex meshes, as it will be detailed in the next sections.

The weights β_i can be calculated using an analogue method to the one used for mesh refinement in (Yang, 2005). We are looking for an interpolated point q at the center of each face. Assuming that points p_i define a face, and $\overrightarrow{N_i}$ are the unit normal vectors to the mesh at p_i, then we can estimate the position for q as:

$$\bar{q} = c_u + w \sum_{i=1}^{L} ((p_i - c_u) \cdot \overrightarrow{N_i}) \overrightarrow{N_i} \tag{13}$$

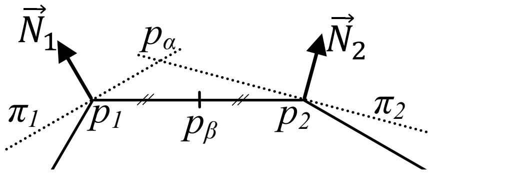


Figure 3: Solution of equation (4) as the affine combination of the generalized intersection of planes π_i (p_α) and the average of all points p_i (p_β, here for $\beta_i = \beta$).

where w is a free positive parameter controlling the smoothness of the interpolation, and where:

$$c_u = \frac{1}{L} \sum_{i=1}^{L} p_i. \tag{14}$$

Replacing q by its estimate $\bar{q} = [\bar{x}, \bar{y}, \bar{z}]^T$ in eq. (11), it follows:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \\ 1 \end{bmatrix} = \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ 1 \end{bmatrix}, \tag{15}$$

Now, the weights β_i that minimize the residues δ should be found, such that $\bar{q}$ approximates the solution of equation (15) for those β_i. Because q should lie close to the face center, the same weight can be assigned to all points, *i.e.* $\beta_i = \beta$. Using the original planes to express the residues δ, it follows:

$$\begin{aligned} \delta_x &= \sum_{i=1}^{L} \alpha_i a_i (A_i \cdot \bar{q}) + \beta \left(L\bar{x} - \sum_{i=1}^{L} x_i \right) \\ \delta_y &= \sum_{i=1}^{L} \alpha_i b_i (A_i \cdot \bar{q}) + \beta \left(L\bar{y} - \sum_{i=1}^{L} y_i \right) \\ \delta_z &= \sum_{i=1}^{L} \alpha_i c_i (A_i \cdot \bar{q}) + \beta \left(L\bar{z} - \sum_{i=1}^{L} z_i \right) \end{aligned} \tag{16}$$

Then, finding the weight β can be achieved by minimizing $\delta_x^2 + \delta_y^2 + \delta_z^2$. The solution of $\partial(\delta_x^2 + \delta_y^2 + \delta_z^2)/\partial\beta = 0$ leads to:

$$\beta = \frac{TB}{B^2} \tag{17}$$

where:

$$\begin{aligned} T &= \sum_{i=1}^{L} \alpha_i (A_i \cdot \bar{q}) \overrightarrow{N_i}, \\ \text{and } B &= \sum_{i=1}^{L} (p_i) - L\bar{q} \end{aligned} \tag{18}$$

4 FROM TRIANGULATION TO SIMPLEX SURFACE MESH

In this section, we will see the first case, *i.e.* converting a triangulation into a simplex surface mesh. In this case, an appropriate vertex q_u on the new simplex mesh must be calculated for each triangular face t_u. Then, we need information for each triangle t_u about the curvature of the mesh. Let us consider the tangent planes to the vertices p_i $(i = 1,2,3)$ composing triangle t_u (Fig. reffig:schemeTri); these planes π_i can be written as $A_i \cdot p = 0$ as defined previously. The normal vectors that define these planes can be calculated as:

$$\overrightarrow{N_i} = \frac{\sum_{k=1}^{L_i} \phi_k \overrightarrow{N_k}}{\left\| \sum_{k=1}^{L_i} \phi_k \overrightarrow{N_k} \right\|}, \tag{19}$$

where $\overrightarrow{N_k}$ $(k = 1, \dots, L_i)$ are the normals of the triangles t_k to which the vertex p_i belongs, and ϕ_k is the angle of the triangle t_k at vertex p_i (Fig. 4).

To approximate the surface, the distance between the new vertex q_u and planes π_i is minimized. Again, q_u should not lie too far from the center of triangle t_u to preserve a regular shape, therefore q_u should minimize its distance to vertices p_i. As aforementioned, the direct minimization of eq. (4) will provide us with an appropriate q_u.

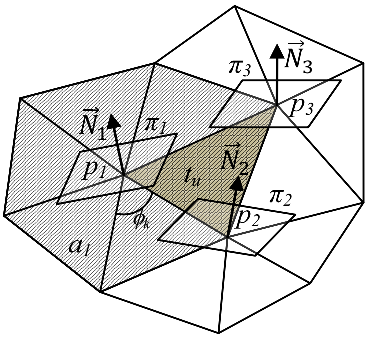


Figure 4: Scheme of triangle t_u, planes and vertices used to find vertex q_u of the dual simplex mesh.

Each weight α_i computation is based on the area a_i corresponding to the sum of the areas of all triangles t_k sharing p_i (Fig. 4):

$$\alpha_i = \frac{a_i}{\sum_{j=1}^{3} a_j}. \tag{20}$$

This way, the distance to each plane is weighted according to the area of triangles that were used to calculate it. The weights β_i are calculated using the same technique as described in section 3, equation (17).

5 FROM SIMPLEX TO TRIANGULATION SURFACE MESH

In this section, we are dealing now with the converse case. A vertex q_u of the triangulation must be calculated for each face f_u of the simplex mesh. However, faces of a simplex mesh do not have a fixed number of vertices p_i $(i = 1, \dots, N_u)$, and moreover they are generally not coplanar. The distance between q_u and the planes π_i tangent to the vertices p_i, is minimized to maintain the geometry of the mesh. These planes are defined by the vertices p_i and the normal vector at each vertex. In a simplex mesh, normals are defined by the plane containing the three neighbors $p_{N1(i)}, p_{N2(i)}, p_{N3(i)}$ (Fig. 2) of the considered vertex p_i (Delingette, 1999). As in the inverse case, q_u should lie close to the center of the face f_u to preserve a regular shape. Figure 5 illustrates these planes and vertices. As previously, eq. (4) can be used to calculate q_u by minimizing the distance to planes π_i and vertices p_i.

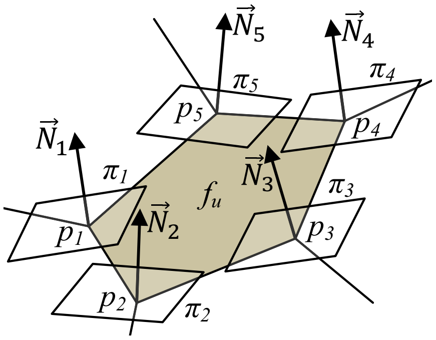


Figure 5: Scheme of face f_u, planes and vertices used to find vertex q_u of the dual triangulation.

The surface of the circle defined by the neighbors of each vertex p_i is a good estimation of the importance the plane π_i has within the mesh, therefore its radius r_i is used to calculate the weights α_i (Fig. 2(a)). It follows:

$$\alpha_i = \frac{r_i^2}{\sum_{j=1}^{N_u} r_j^2} \tag{21}$$

Again, in this case, weights β_i are calculated using the same technique described in section 3, equation (17).

6 RESULTS

When performing a transformation between simplex meshes and triangulations (and conversely), a similar mesh to the original one is expected, in order to result in a minimal geometric perturbation. To measure the quality of the transformations in both directions, the set of successive transformations $(T_1 \rightarrow S_1 \rightarrow T_2 \rightarrow \cdots \rightarrow T_k \rightarrow S_k \rightarrow T_{k+1} \rightarrow \cdots \rightarrow T_N \rightarrow S_N)$ is performed, where T_k is a triangulation and S_k a simplex mesh, with $(k = 1, \dots, N)$. It is obvious that such back and forth conversion will never be required by any application, but successive transformations permit to magnify, and thus pointing out, incorrect behaviors of a technique.

The proposed technique has been compared to the most commonly used at this time, *i.e.* using the Center of Mass of each face to compute the corresponding vertex of the dual mesh (Delingette, 1999). Since all meshes T_k have the same number of vertices, idem as do the S_k between them, we have considered that the most appropriate measure was a simple vertex-to-vertex distance computation after each transformation cycle. This way, each triangulation is compared at each step to the initial triangulation; and correspondingly, each simplex meshes is considered accordingly to the first simplex mesh obtained.

Figure 6 shows the distance graph measured for the surface of cerebral ventricles (1360 vertices/simplex faces, 2728 triangles/simplex vertices), for 150 iterations. The vertex-to-vertex mean distances are expressed as a percentage of the bounding box diagonal of T_1 or S_1, respectively. Curve 6(a) shows results using the *Center of Mass* technique, while 6(b) draws results with our original technique. If we compare the results for a set of meshes, the *Center of Mass* technique produces high degeneration in some parts of the mesh (Fig. 7(b), (d) and (f)), losing most of the details present in the initial geometry. However, using an interpolation based on the tangent planes as presented in this article, it can be clearly seen on Fig. 7(c), (e) and (g), that the initial geometry is much better preserved.

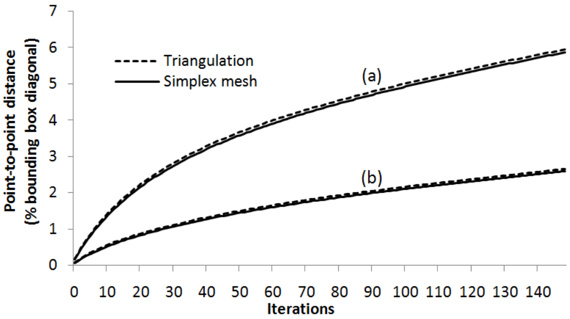


Figure 6: Curves of the mean error of the successive transformations of a cerebral ventricles surface (a) Transformation based on the faces center of mass. (b) Interpolation based on tangent planes.

As a complementary result, the Hausdorff distance was measured as well between initial and transformed meshes by using the `Metro` tool that adopts a surface sampling approach (Cignoni et al., 1998). The *Prism* (92 vertices, 180 triangles; *from AIM@SHAPE*), *Block*(2132 vertices, 4272 triangles; from *AIM@SHAPE*), *Horse* (48485 vertices, 96966 triangles; from *Cyberware, Inc*), and *Bunny* (34834 vertices, 69451 triangles; from *Stanford 3D Scanning Repository*) meshes have been considered; and the distance was measured after a cycle of transformations, *i.e.* swapping back and forth to simplex mesh and triangulation (once again, this is totally artificial and is achieved for testing purposes only, considering that our method is local and direct, and that generally a single one-way conversion is wanted). Figure 8 shows the initial meshes with coloration according to their distance to the resulting one, and Table 1 shows the well known ratio between measured distances and the bounding box diagonal of the original mesh. The mean and Root Mean Square (RMS) distances between two surfaces M_1 and M_2 are defined as:

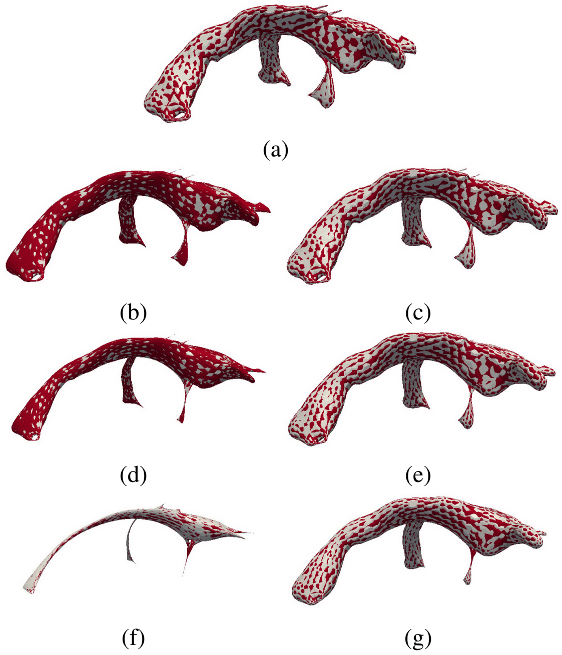


Figure 7: Cerebral ventricles mesh after successive transformations between simplex (lighter) mesh and triangulation (darker). Left: meshes obtained using the faces' mass centers, after (b) 5, (d) 15 and (f) 50 cycles. Right: meshes obtained using our technique, after (c) 5, (e) 15 and (g) 50 cycles.

$$\text{Mean dist.}(M_1,M_2) = \frac{1}{|M_1|}\int_{p\in M_1} \text{HD}(p,M_2)ds$$

$$\text{RMS dist.}(M_1,M_2) = \sqrt{\frac{1}{|M_1|}\int_{p\in M_1} \text{HD}(p,M_2)^2 ds},$$

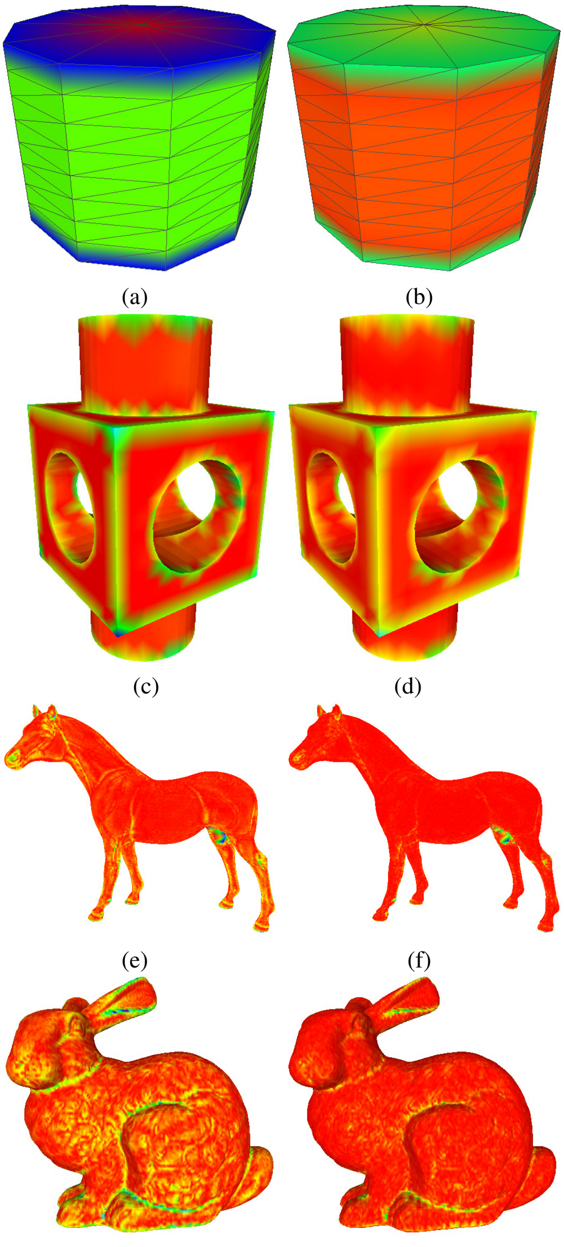
where $\text{HD}(p,M)$ is the Hausdorff distance between point p and surface M, and $|M|$ its area. As it can be guessed, in both cases, the main error is concentrated in high curvature areas. But, as previously seen, the error dramatically decreases with our technique (Fig. 8, right column) compared to the *Center of Mass* (left column).

Figure 9 shows a comparison between the initial (darker) and the resulting (lighter) meshes, using both methods. Errors are significantly lower in our case (b) than for the *Center of Mass* technique (a). Moreover, the resulting mesh tends to be inside (resp. outside) the initial mesh in areas with positive (resp. negative) curvature for the classic technique, while our technique avoids this construction artifact, thanks to the introduction of an appropriate weighting between element regularity and surface smoothness. Moreover, from examining equation (4), the question of the topological validity of the resulting mesh may arise. The solution is an equilibrium between shape preservation and mesh smoothing, that behaves properly (*i.e.* the point lays inside the triangle). However, for extreme cases like spiky meshes with very high curvature areas, some additional feature preserving process may be required.

The computation time was multiplied by approximately 30 with our method; *e.g.* the computation time for the prism mesh was 7 milliseconds with the center of mass and 270 milliseconds with our method[1]. However, our method is direct and performed locally for each vertex, thus computational time is linear according to the number of vertices of the mesh. And from the results we obtain, we believe it is worth paying an extra (but limited) amount of computation to drastically improve the final quality of the dual mesh.

Table 1: Hausdorff distances.

		Center of Mass	Distance to Planes	Gain [%]
Prism Mesh	min	0,003537	0,000016	99,54
	max	0.060099	0.037205	38.09
	mean	0.033701	0.014088	58.20
	RMS	0.036620	0,018715	48,89
Block Mesh	min	0.0	0.0	0.0
	max	0.019153	0.014321	25.23
	mean	0.002397	0.001820	24.07
	RMS	0.003855	0.002840	26.34
Horse Mesh	min	0.0	0.0	0.0
	max	0.004596	0.003873	15.74
	mean	0.000126	0.000047	62.50
	RMS	0.000205	0.000107	48.08
Bunny Mesh	min	0.0	0.0	0.0
	max	0.003321	0.002761	16.85
	mean	0.000220	0.000096	56.36
	RMS	0.000324	0.000160	50.62



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 8: *Prism*, *Block*, *Horse* and *Bunny* meshes colored according to the Hausdorff distance after a cycle of transformations. 1) Left, subfigures a), c), e) and g) using *Center of Mass*. 2) Right, subfigures b), d), f) and h) using our method based on *Distance to the tangent planes*.

7 CONCLUSION AND DISCUSSION

We have presented a method to carry out transformations between triangulations and simplex meshes, and vice-versa. Compared to the ones proposed in the lit-

[1] developed in Python Language on AMD Athlon 62x2 Dual, 2GHz, 1Gb RAM

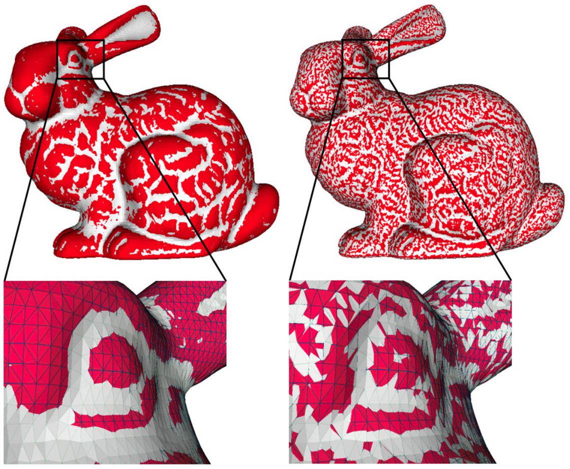

(a) (b)

Figure 9: Comparison between the *bunny* original mesh (darker) and after a cycle of transformations (lighter). (a) Using *Center of Mass*. (b) Using our distance to the tangent planes.

erature, our method is straightforward and does not require any iteration. It is intuitively based on the interpolation of the initial mesh to find the corresponding vertices of the dual mesh. The interpolation is based on a direct and local minimization of the distance to tangent planes, and vertices of each face. Our transformation technique was compared to the most frequently used method, which is based on placing the dual vertices at the center of mass of the initial faces, and the weaknesses of this latter have been illustrated. The performance of the proposed method was measured using a vertex-to-vertex distance between both triangulations and simplex meshes, after performing a chain of successive transformations. Moreover, we measured the Hausdorff distance between meshes after performing a cycle of transformations, *i.e.* after carrying out a transformation to simplex mesh and back to triangulation. The performance of our method was more than satisfactory, providing a more than significant reduction of the error, of nearly 50%, at reasonable linear time. Thus, our method has proven to be adequate to be used in any application requiring topological mesh transformation while preserving geometry, and without increasing complexity.

REFERENCES

Alliez, P., Ucelli, G., Gotsman, C., and Attene, M. (2008). Recent advances in remeshing of surfaces. In *Shape Analysis and Structuring*, Mathematics and Visualization, pages 53–82. Springer Berlin Heidelberg.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22(4):469–483.

Cignoni, P., Rocchini, C., and Scopigno, R. (1998). Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174.

de Putter, S., Vosse, F., Gerritsen, F., Laffargue, F., and Breeuwer, M. (2006). Computational mesh generation for vascular structures with deformable surfaces. *Int. J. of Computer Assisted Radiology and Surgery*, 1:39–49.

Delingette, H. (1999). General object reconstruction based on simplex meshes. *Int. J. of Computer Vision*, 32(2):111–146.

Garland, M. and Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In *ACM SIGGRAPH proceedings*, pages 209–216.

Heckbert, P. S. and Garland, M. (1999). Optimal triangulation and quadric-based surface simplification. *Computational Geometry: Theory and Applications*, 14(1-3):49 – 65.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high-resolution 3d surface construction algorithm. *ACM Computer Graphics*, 21(4):163–170.

Owen, S. J. (1998). A survey of unstructured mesh generation technology. In *Int. Meshing Roundtable*, pages 239–267.

Ronfard, R. and Rossignac, J. (1996). Full-range approximation of triangulated polyhedra. *Proc. of Eurographics, Computer Graphics Forum*, 15(3):67–76.

Si, H. (2010). Constrained delaunay tetrahedral mesh generation and refinement. *Finite Elements in Analysis and Design*, 46(1-2):33–46.

Treece, G. M., Prager, R. W., and Gee, A. H. (1999). Regularised marching tetrahedra: improved iso-surface extraction. *Computer & Graphics*, 23(4):583–598.

Xu, C., Pham, D. L., Rettmann, M. E., Yu, D. N., and Prince, J. L. (1999). Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Trans. on Medical Imaging*, 18(6):467–480.

Yang, X. (2005). Surface interpolation of meshes by geometric subdivision. *Computer-Aided Design*, 37(5):497–508.