# 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, Lyon, France*
[3]*Université de Lyon, CNRS, Université Lyon 1, LIRIS, SAARA team, UMR5205, F-69622, Lyon, France*

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

Abstract: We propose an accurate method to convert from a triangular mesh model to a simplex mesh and vice-versa. 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. Our method is based on the distance minimization between the local tangent planes of the mesh and vertices of each face.

# 1 INTRODUCTION

Deformable model techniques are widely used in image segmentation tasks. Among these models, simplex meshes are valuable candidates (Delingette, 1999), for their favorable characteristics in this type of modeling, as its convenient way to model internal forces. With this type of meshes, as with triangulations, any topology can be described. Triangulations are meshes of triangles in which, if they are manifolds, each triangle has three neighboring triangles, while simplex meshes are meshes of polygons in which each vertex has three neighboring vertices. Simplex meshes and triangulations are topologically duals (Delingette, 1999), and this allows us to naturally obtain a simplex mesh by applying a dual operation to the triangulation, and vice-versa. Moreover, there are some tasks for which simplex meshes are not suitable, and thus triangulated meshes are more adapted. Examples of these applications are: generation of volumetric meshes, rendering and calculation of area.

Currently, the most common way to perform conversions between triangulations and simplex meshes is to determine the set of vertices for the final mesh as the gravity center of each face of the initial mesh (*e.g.* (The Insight Segmentation and Registration Toolkit (ITK), 2011)). This technique is fast, but unfortunately in this case, mesh smoothing is generally high; original shape (curvature) and volume is not properly respected. In (De Putter et al., 2006), an iterative curvature correction algorithm for the dual triangulation of a two-simplex mesh is proposed. Their solution provides 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.

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 finding an interpolation of the center of each face, and to build the dual mesh accordingly to these points. 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 perform triangular mesh simplifications. Another equivalent measure has been employed, using this time a summation to obtain a quadratic error in (Garland and Heckbert, 1997; Heckbert and Garland, 1999). In a more recent work, a method for refining triangulations has been developed (Yang, 2005) using a similar measure. It is worth pointing out that our global objective is to perform a transformation between meshes, and not to refine them. However, we mainly got inspiration from

J. Galdames F. and Jaillet F..
FROM TRIANGULATION TO SIMPLEX MESH AND VICE-VERSA - A Simple and Efficient Conversion.
DOI: 10.5220/0003851801510156
In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (GRAPP-2012), pages 151-156
ISBN: 978-989-8565-02-0

this last work.

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

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:

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

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

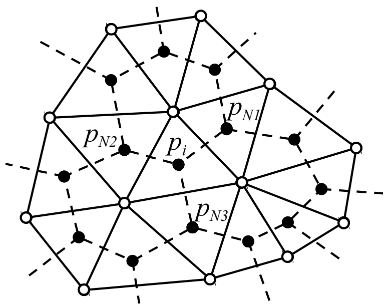



Figure 1: Simplex meshes and triangulations are topological but not geometrical duals. White dots: triangulation vertices; Black dots: simplex mesh vertices. A vertex of the simplex mesh $p_i$ and its three neighbors $p_{N1}, p_{N2}, p_{N3}$ are shown.

Simplex meshes can be used to efficiently implement segmentation methods based on deformable models. Each vertex of a simplex mesh has three neighbors $p_{N1(i)}, p_{N2(i)}, p_{N3(i)}$ (Fig. 1); between them, a restricted number of entities is defined, the simplex angle and the metric parameters (Delingette, 1999). The mesh deformation can be controlled by using these entities.

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 $\pi_i$ 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. A method to achieve the aforementioned goal is explained in the next section.



## 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, accordingly $a^2+b^2+c^2 = 1$, 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 $\pi_i$ represented by $A_i \cdot p = 0$ $(i = 1, \ldots, L)$, the distance between any point $q = [x,y,z,1]^T$ and each plane $\pi_i$ is $|A_i \cdot q|$. On the other hand, consider a set of points $p_j$ $(j = 1, \ldots, M)$. If we want to find the point $q$ minimizing its distance to planes $\pi_i$ and points $p_j$, the function to be considered follows:

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

where $\alpha_i$ and $\beta_j$ are positive weights for the distance to the planes (in order to respect geometry and curvature) and points (controlling shape regularity), respectively. Equation (2) can be rewritten in matrix form as:

$$D(q) = q^T Q q \tag{3}$$

where

$$Q = \sum_{i=1}^{L} \alpha_i A_i^T A_i + \sum_{j=1}^{M} \beta_j Q_j \tag{4}$$

and

$$Q_j = \begin{bmatrix} 1 & 0 & 0 & -x_j \\ 0 & 1 & 0 & -y_j \\ 0 & 0 & 1 & -z_j \\ -x_j & -y_j & -z_j & x_j^2+y_j^2+z_j^2 \end{bmatrix} \quad (5)$$

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

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \quad (6)$$

To minimize the quadratic form of eq. (3), the partial derivatives of $D(q)$ are used: $\frac{\partial D(q)}{\partial x} = 0$, $\frac{\partial D(q)}{\partial y} = 0$, $\frac{\partial D(q)}{\partial z} = 0$. This system of equations can be rewritten in a matrix form as:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

Finally, the solution of eq. (7) follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{12} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix}^{-1} \begin{bmatrix} -q_{14} \\ -q_{24} \\ -q_{34} \end{bmatrix} \quad (8)$$

where $q = [x, y, z]^T$. This system always has a unique solution (*i.e.* the matrix is invertible) because function (2) is strictly convex and therefore has no more than one minimum.

## 3.1 Weights Calculation

The solution of equation (2) can be understood as an affine combination of the generalized intersection of all planes $\pi_i$ (first term) and the average of all points $p_j$ (second term). This affine combination is controlled by the weights $\alpha_i$ and $\beta_i$. For example, let's consider points $p_1, p_2$ and planes $\pi_1, \pi_2$ as shown on Figure 2. Planes intersect at point $p_\alpha$, and the average of the points (for $\beta_i = \beta$) is $p_\beta$. The weights $\alpha_i$ 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 $\beta_i$ can be calculated using a similar 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 $L$ vertices $p_i$ 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:

$$\bar{q} = c_u + w \sum_{i=1}^{L} ((p_i - c_u) \cdot \overrightarrow{N_i}) \overrightarrow{N_i} \quad (9)$$

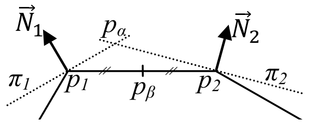



Figure 2: Solution of equation (2) as the affine combination of the generalized intersection of planes $\pi_i$ ($p_\alpha$) and the average of all points $p_i$ ($p_\beta$, here for $\beta_i = \beta$).

where $w$ is a free positive parameter controlling the smoothness of the interpolation, and $c_u$ is the average position of the vertices $p_i$. Replacing $q$ by its estimate $\bar{q} = [\bar{x}, \bar{y}, \bar{z}]^T$ in eq. (7), it follows:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \\ 1 \end{bmatrix} = \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ 1 \end{bmatrix}, \quad (10)$$

Now, the weights $\beta_i$ that minimize the residues $\delta$ should be found, such that $\bar{q}$ approximates the solution of equation (10) for those $\beta_i$. 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 $\delta$, it follows:

$$\begin{aligned} \delta_x &= \sum_{i=1}^{L} \alpha_i a_i (A_i \cdot \bar{q}) + \beta \left( L\bar{x} - \sum_{i=1}^{L} x_i \right) \\ \delta_y &= \sum_{i=1}^{L} \alpha_i b_i (A_i \cdot \bar{q}) + \beta \left( L\bar{y} - \sum_{i=1}^{L} y_i \right) \\ \delta_z &= \sum_{i=1}^{L} \alpha_i c_i (A_i \cdot \bar{q}) + \beta \left( L\bar{z} - \sum_{i=1}^{L} z_i \right) \end{aligned} \quad (11)$$

Then, finding the weight $\beta$ 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:

$$\beta = \frac{TB}{B^2} \quad (12)$$

where:

$$T = \sum_{i=1}^{L} \alpha_i (A_i \cdot \bar{q}) \overrightarrow{N_i}, \qquad B = \sum_{i=1}^{L} (p_i) - L\bar{q} \quad (13)$$

To avoid a negative or zero value of $\beta$, and to keep a regular surface, $\beta = \min(\max(TB/B^2, 0.1), 2)$

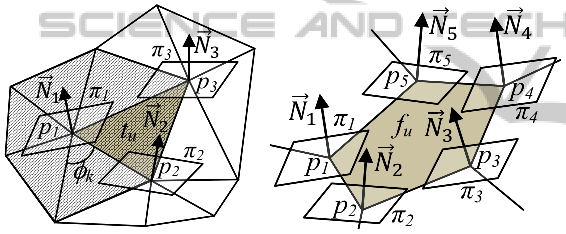
# 4 FROM TRIANGULATION TO SIMPLEX SURFACE MESH

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 computed 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. 3(a)); these planes $\pi_i$ can be written as $A_i \cdot p = 0$ as defined previously. The normal vectors that define these planes can be calculated as:

$$\overrightarrow{N_i} = \frac{\sum_{k=1}^{L_i} \phi_k \overrightarrow{N_k}}{\left\| \sum_{k=1}^{L_i} \phi_k \overrightarrow{N_k} \right\|}, \tag{14}$$

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 $\phi_k$ is the angle of the triangle $t_k$ at vertex $p_i$ (Fig. 3(a)).

To approximate the surface, the distance between the new vertex $q_u$ and planes $\pi_i$ 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 $p_i$. The direct minimization of eq. (2) will provide us with an appropriate $q_u$.



(a) (b)

Figure 3: (a) Scheme of triangle $t_u$, planes and vertices used to find vertex $q_u$ of the dual simplex mesh. (b) Scheme of face $f_u$, planes and vertices used to find vertex $q_u$ of the dual triangulation.

Each weight $\alpha_i$ computation is based on the area $a_i$ corresponding to the sum of the areas of all triangles $t_k$ sharing $p_i$ (Fig. 3(a)):

$$\alpha_i = \frac{a_i}{\sum_{j=1}^{3} a_j}. \tag{15}$$

This way, the distance to each plane is weighted according to the area of triangles that were used to calculate it. The weights $\beta_i$ are calculated with equation (12).

# 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 $\pi_i$ 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 $p_{N1(i)}, p_{N2(i)}, p_{N3(i)}$ 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 3(b) illustrates these planes and vertices. As previously, eq. (2) can be used to calculate $q_u$ by minimizing the distance to planes $\pi_i$ and vertices $p_i$.

The surface of the circle defined by the neighbors of each vertex $p_i$ is a good estimation of the importance the plane $\pi_i$ has within the mesh, therefore its radius $r_i$ is used to calculate the weights $\alpha_i$. It follows:

$$\alpha_i = \frac{r_i^2}{\sum_{j=1}^{N_u} r_j^2} \tag{16}$$

Again, in this case, weights $\beta_i$ are calculated with equation (12).

# 6 RESULTS

To measure the quality of the transformations in both directions, the set of successive transformations $(T_1 \to S_1 \to T_2 \to \cdots \to T_k \to S_k \to T_{k+1} \to \cdots \to T_N \to S_N)$ 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 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 $T_k$ and $S_k$ have respectively the same number of vertices, 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 4 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

Table 1: Hausdorff distances and computational time.

| | | Center of Mass | Distance to Planes | Gain [%] |
|---|---|---|---|---|
| Block Mesh | min | 0.0 | 0.0 | 0.0 |
| | max | 0.019153 | 0.014321 | 25.23 |
| | mean | 0.002397 | 0.001820 | 24.07 |
| | time [ms] | 165.219 | 6330.415 | -3731.53 |
| Horse Mesh | min | 0.0 | 0.0 | 0.0 |
| | max | 0.004596 | 0.003873 | 15.74 |
| | mean | 0.000126 | 0.000047 | 62.50 |
| | time [ms] | 3718.78 | 116221.5 | -3025.26 |
| Bunny Mesh | min | 0.0 | 0.0 | 0.0 |
| | max | 0.003321 | 0.002761 | 16.85 |
| | mean | 0.000220 | 0.000096 | 56.36 |
| | time [ms] | 2734.51 | 86895.5 | -3077.74 |

box diagonal of $T_1$ or $S_1$, respectively. Curve 4(a) shows results using the *Center of Mass* technique, while 4(b) draws results with our 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. 5(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. 5(c), (e) and (g), that the initial geometry is much better preserved.

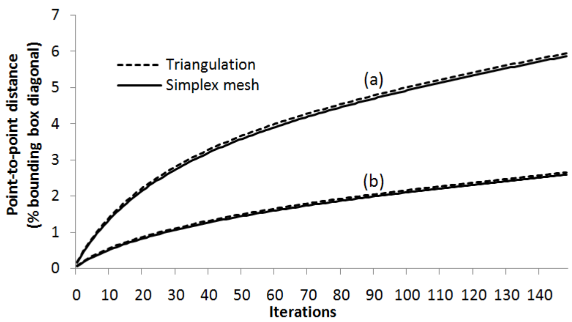



Figure 4: Curves of the mean error of the successive transformations of a cerebral ventricles surface (a) Transformation based on the faces center of mass. (b) Interpolation based on tangent planes.

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 *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. Figure 6 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 distance between two surfaces $M_1$ and $M_2$ is defined as: Mean dist.$(M_1, M_2) = \frac{1}{|M_1|}\int_{p \in M_1} \mathrm{HD}(p, M_2)ds$, where $\mathrm{HD}(p, M)$ 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. 6, right column).

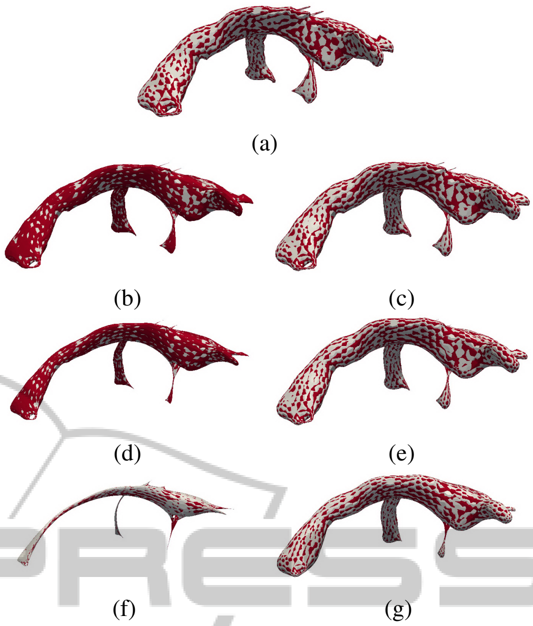



Figure 5: Cerebral ventricles mesh (a) after successive transformations between simplex (lighter) mesh and triangulation (darker). Left: meshes obtained using the faces' mass centers, after (b) 5, (d) 15 and (f) 50 cycles. Right: meshes obtained using our technique, after (c) 5, (e) 15 and (g) 50 cycles.

From examining equation (2), 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 high curvature areas, some additional feature preserving process may be required.

Table 1 also shows the computational time of each mesh in milliseconds[1]. The computational time was

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

multiplied by approximately 33 with our method. The computational time is linear according to the number of vertices of the mesh because our method is direct and performed locally for each vertex. 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. Moreover, the large computational time may be explained by the fact that our implementation is carried out in an interpreted language (Python). By performing a simple test of numerical computation, we have verified that the computational time can be reduced about 400 times with a C++ implementation instead of Python.

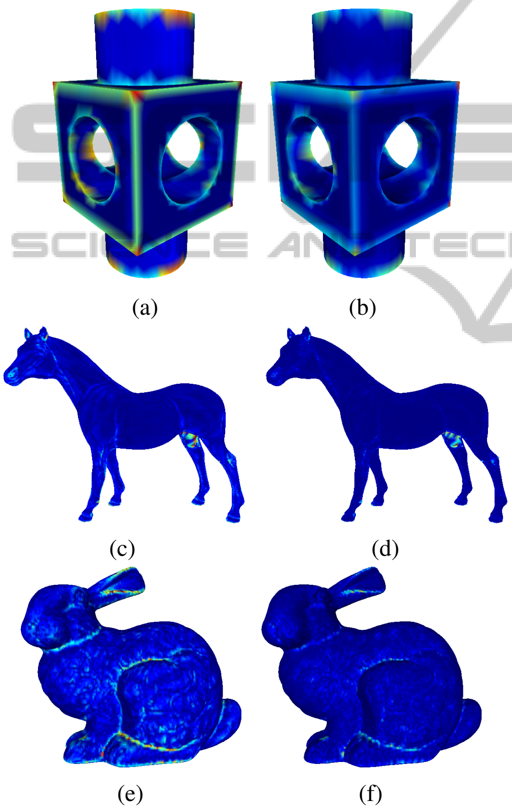


Figure 6: *Block*, *Horse* and *Bunny* meshes colored according to the Hausdorff distance after a cycle of transformations. The blue-green-red color scale is used, where red represents the largest value .1) Left, subfigures (a), (c) and (e) using *Center of Mass*. 2) Right, subfigures (b), (d) and (f) using our method based on *Distance to the tangent planes*.

## 7 CONCLUSIONS 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 literature, 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 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


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.

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

The Insight Segmentation and Registration Toolkit (ITK) (2011). www.itk.org. www.itk.org.

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