

# ABOUT SURFACE REMESHING

Pascal J. FREY,

*INRIA, Gamma Project,  
Domaine de Voluceau, Rocquencourt,  
BP 105, 78153 Le Chesnay Cedex, France.  
Email: Pascal.Frey@inria.fr*

## ABSTRACT

In this paper, we present a general scheme suitable to optimize an arbitrary given surface triangulation representing a piecewise linear approximation of an underlying surface geometry. The proposed approach is based on two steps related to i) the extraction of a geometric mesh (based on a simplification procedure), ii) the optimization of this mesh with respect to the model geometry as well as to the element shape quality (for finite element computations). This last stage involves the creation of a geometric ( $G^1$  continuous) support that defines an adequate approximation of the underlying surface and the definition of a geometric (curvature-based) metric. Details about the construction of the metric and the definition of a proper simplification “enveloppe” are provided. Examples of surface mesh simplification are given to illustrate the various stages of the proposed approach.

**Keywords:** Surface remeshing, Triangulation, Hausdorff distance, Decimation, Mesh adaptation.

## 1. INTRODUCTION

Surface remeshing has become very important today as it is now possible to handle large models (with great level of details or very complex geometries). Hence, large meshes are commonly generated in numerous applications areas. For instance, in computer graphics or geometric modelling, large polygonal meshes are produced as piecewise linear approximations of curved parametric surfaces. Using modern range scanning devices, it is possible to acquire terrains or models consisting of millions of polygons. On the other hand, in finite element analysis, the quality of the surface approximation is also important as it may affect the accuracy of the numerical results.

In this context, the correct and accurate representation of surfaces is essential. Usually, surfaces can

be defined in two ways, either using a set of conforming patches (for instance, in CAD systems) or in a discrete manner (*i.e.*, a surface triangulation, for instance obtained using a reconstruction algorithm based on a set of points provided by a scanning device). In the latter, the number of elements in the initial surface triangulation can be rather large (e.g. several millions for biomedical data). Irrespective of the domain of application, too many mesh elements is generally considered as a penalty and may impede any further computation based on this mesh. It is thus necessary to reduce this number to a reasonable level, while preserving the accuracy of the geometric approximation as well as the mesh quality.

For surface meshes, the validity of the geometric approximation within a given tolerance is related to the control of the distance between the mesh

elements and the underlying surface. Hence, a *geometric mesh* [6] is a mesh such that : all vertices belong to the surface, all elements are close to the surface and any element is close to the tangent plane at its surface vertices. The first two requirements enable to bound the distance (gap) between the mesh and the surface (surface deviation), as the third constraint is introduced to ensure a local  $G^1$  continuity of the surface (surface smoothness).

In addition to these requirements, for numerical simulations based on finite or boundary element methods, additional constraints need to be taken into account. For instance, the mesh elements must be well-shaped (equilateral triangles) as badly shaped elements may alter the accuracy of the numerical results [4].

In this paper, we present a surface remeshing scheme suitable to obtain geometric as well as finite element meshes given an initial surface triangulation  $T$ . The whole process involves three main stages. At first, the triangulation  $T$  is analyzed in order to construct a geometric mesh  $M_G$  in which the distance from the initial approximation is bounded. This mesh can be considered as an accurate piecewise linear approximation of the underlying surface, although it may not be suitable for numerical computations. Then, depending on the context of application, the mesh  $M_G$  is optimized so as to improve the element shape quality and/or to enforce prescribed element sizes. To this end, a  $G^1$  continuous geometric support is defined on  $M_G$ , that will be used to construct a geometric metric  $\mathcal{G}_3$  (defined in the tangent planes) as well as to govern the mesh modification operations. Finally, the mesh  $M_G$  is locally optimized to obtain a *unit surface mesh*  $M$  conforming the metric  $\mathcal{G}_3$  (possibly coupled to another metric  $\mathcal{M}_2$ , for instance provided by an error estimate).

The construction of a geometric mesh from a given surface triangulation is discussed in Section 2. The numerical approximation of the principal curvatures at mesh vertices is discussed in Section 3 as well as the definition of a geometric metric. The construction of a geometric support is described in Section 4. In Section 5, the extraction of a *unit surface mesh* is detailed. Finally, application examples are provided in Section 6 to illustrate the proposed scheme.

## 2. GEOMETRIC SURFACE MESH

The aim of this stage is to transform the initial triangulation into a geometric surface mesh containing less elements, if possible, in accordance to the geometric requirements.

The approach we suggest here guarantees that the elements of the simplified mesh stay within a given (user-specified) tolerance of the original surface triangulation. The tolerance region can be seen as an *envelope* that surrounds the initial triangulation, in which the simplification is performed [5].

### 2.1 Definition of the envelope

Usually, we consider a tolerance value that bounds the maximum allowable deviation of the simplified mesh from the initial one. The goal is to compute a piecewise linear approximation  $M$  of the initial triangulation  $T$  such that every point of  $T$  is within a distance  $\delta$  of a point of  $M$  and, conversely, that every point of  $M$  is within a distance  $\delta$  of some point of  $T$ . As it appears, this geometric deviation can be related to the Hausdorff distance as the latter represents the maximum distance of a set to the nearest point in the other set.

**Hausdorff distance.** Formally speaking, the one-sided Hausdorff distance from a set  $M$  to set  $T$  is a *maximin* function, defined as :

$$h(M, T) = \max_{\{P \in M\}} \min_{\{Q \in T\}} d(M, T), \quad (1)$$

where  $P$  and  $Q$  are points of sets  $M$  and  $T$  respectively, and  $d(.,.)$  is any *metric* between these points (we consider here the usual Euclidean distance). Notice that it does not define a distance function because it is not symmetric. One way to overcome this problem is thus to consider the (full) Hausdorff distance between two sets  $M$  and  $T$  :

$$H(M, T) = \max(h(M, T), h(T, M)). \quad (2)$$

Here,  $T$  being the initial triangulation, the resulting geometric mesh  $M = \bigcup_j K_j$  must be such that :

$$H(K_j, T) \leq \delta, \quad (3)$$

where  $\delta$  denotes the maximum (user-specified) allowable distance between the two surface approximations. Practically, the main difficulty consists

in computing the Hausdorff distance (which is expensive to compute exactly).

Unfortunately, computing the exact Hausdorff distance between the initial and the simplified mesh is a rather complex and tedious task, because not only the distances between vertices have to be computed but also the distances between all points of the two meshes. For the sake of simplicity and efficiency, we will not compute the exact two-sided Hausdorff distance, but rather a sharp majorant of this value.

**Evaluation of the deviation.** A straightforward approach consists in computing and accumulating the error during the simplification process [18]. The deviation is computed at stage  $i$ , when mesh  $M_i$  replace mesh  $M_{i-1}$  and this value is added to the previous deviation <sup>1</sup>.

To this end, we consider here that one mesh vertex is removed at the time by an edge collapsing operation. This allows us to restrict the region of interest (affected by the operation) to the ball of the vertex to be removed (*i.e.*, the set of triangles sharing this vertex).

Let consider the ball  $\mathcal{B}^i(P) = \{K_j^i(P) \in M_i\}$  of a vertex  $P$  in the current mesh  $M_i$ . The necessary condition to preserve the geometric approximation of each triangle  $K_j^i \in M_i$  within the given tolerance  $\delta$  is such that  $H(K_j^i, M) \leq \delta$ . This leads to the following relation :

$$H(K_j^i(Q), M_{i-1}) \leq \max(H(K_j^i(Q), \mathcal{B}^{i-1}(P)), H(K_j^{i-1}(P), \mathcal{B}^i(Q))). \quad (4)$$

For the sake of clarity, we focus here on the one-sided distance from a triangle  $K_j^i$  of the new mesh  $M_i$  to the ball of a vertex in mesh  $M_{i-1}$ . As we are dealing with straight-sided (linear) triangles, the distance  $H(K_j^i(Q), \mathcal{B}^{i-1}(P))$  can be evaluated explicitly. More importantly, let us notice that each triangle of mesh  $M_i$  shares a common edge with a triangle of mesh  $M_{i-1}$  (because of the collapsing operator used). Hence, we only distinguish two cases :

- i) the orthogonal projection  $\pi(Q)$  of point  $Q$  with respect to  $K_j^i(Q)$  belongs to a single triangle  $K_j^{i-1}(P)$ ,

<sup>1</sup>It is also possible to accumulate a quadric error instead of the distance [9].

- ii) the triangle  $K_j^i(Q)$  intersects more than one triangle of  $\mathcal{B}^{i-1}(P)$ .

In case *i*) (Figure 1), the distance of triangle  $K_j^i(Q) = ABQ$  to the mesh  $M_{i-1}$  is given by the (orthogonal) projection of point  $Q$  onto the plane  $(ABP)$  :

$$\delta_1 = H(K_j^i(Q), \mathcal{B}^{i-1}(P)) = d(Q, (ABP)).$$

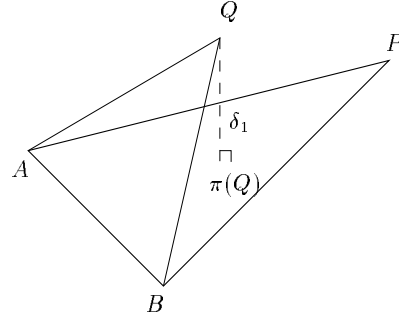


Figure 1. Evaluation of the geometric deviation between two meshes when the orthogonal projection of point  $Q \in M^i$  belongs to a triangle of  $M^{i-1}$ .

In case *ii*) (Figure 2), the evaluation involves the points  $I_k$ , intersections of the edges of the  $\pi(\mathcal{B}^i(Q))$  with the edges of the triangles  $K_j^i(Q)$  (where  $\pi(\mathcal{B}^i(Q))$  denotes the projection of the ball of  $Q$  onto  $\mathcal{B}^{i-1}(P)$ ). As pointed out, the evaluation of the deviation involves several regions (3 on the Figure). The desired distance is thus the maximum distance between the points  $I_k$  and the

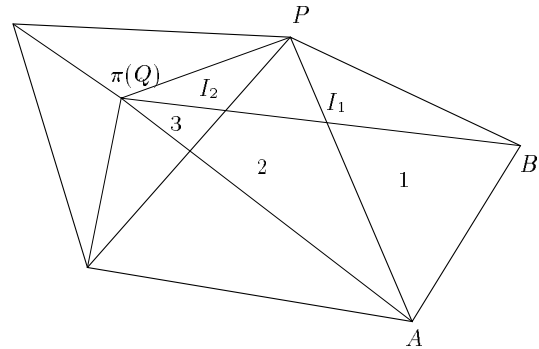


Figure 2. Evaluation of the geometric deviation between two meshes when a triangle  $K_j^i \in M^i$  overlaps three triangles  $K_k^{i-1} \in M^{i-1}$ ,  $k = 1, 3$ .

points  $\pi^{-1}(I_k)$  (on the edges  $P_kQ$ ). More precisely :

$$\delta_2 = \max_k d(\pi(I_k), I_k).$$

As a full Hausdorff distance is sought, we also have to take into account the distance from  $P$  to the projection of  $P$  to the plane of a triangle  $K_j^i(Q) = ABQ$ . Finally, the evaluation formula is :

$$H(K_j^i(Q), \mathcal{B}^{i-1}(P)) = \max(\delta_2, \delta_3), \quad (5)$$

$$\text{where } \delta_3 = \begin{cases} d(P, \pi(P)) & \text{if } \pi(P) \in K_j^i(Q) \\ 0 & \text{otherwise.} \end{cases}$$

Note that this computation provides the full Hausdorff distance between mesh  $i$  and mesh  $i-1$ . To achieve an evaluation of the total deviation (between the current mesh  $M_i$  and the initial surface triangulation  $M_0$ ), the current Hausdorff distance  $H(K_j^i(Q), \mathcal{B}^{i-1}(P))$  is added to each triangle in each iteration step. This leads to a majoration (upper bound)  $b_i$  of its Hausdorff distance to the initial mesh :

$$b_i(K_j^i) = b_{i-1}(K^{i-1}) + H(K_j^i(Q), \mathcal{B}(P)).$$

Finally, the deviation of a triangle  $K_j^i(P)$  will be acceptable if :  $b_i(K_j^i) \leq \delta$ .

## 2.2 Simplification algorithm

Once this deviation computation algorithm has been implemented, the geometric simplification algorithm becomes straightforward. It is essentially based on an iterative edge collapsing procedure coupled with (coplanar) edge swapping operations.

As realistic meshes involves ridges, corners and curves traced onto the surface, a pre-processing stage consists in identifying the  $C^1$  discontinuities of the model (corners, sharp edges, etc.) and in computing the normals at the mesh vertices.

**Quality measure.** In order to control the quality degradation during the collapse operation, we use the finite element shape quality measure :

$$Q(K) = \alpha \frac{h_{max}}{\rho_K}, \quad (6)$$

where  $\rho_K$  denotes the inradius of  $K$ ,  $H_{max}$  is the largest edge length and  $\alpha$  is a normalization coefficient so that  $Q(K) = 1$  for an equilateral triangle. This monotonous shape quality function

ranges from 1 (equilateral triangle) to  $\infty$  (null triangle).

This measure is used to control the degradation of the mesh quality during collapsing operations. Hence, an vertex will be removed provided that :

$$Q(K_j^i(Q)) \leq \omega \max_j Q(K_j^{i-1}(P)),$$

where  $\omega$  is a coefficient enabling to bound the quality degradation.

**Control of the surface roughness.** As indicated in the introduction, in a geometric mesh, each triangle must be close to the tangent plane at its surface vertices. To check whether or not an element  $K_j^i$  conforms this smoothness requirement, we use the following measure :

$$\min_j \langle \vec{\nu}(K^i(P_j)), \vec{n}(K^i) \rangle \leq \cos \theta,$$

where  $\vec{\nu}(K^i(P_j))$  denotes the normal to the surface at vertex  $P_j$ ,  $\vec{n}(K^i)$  is the unit normal to  $K_j^i$  and  $\cos \theta$  is a given tolerance value (in practice,  $\theta \approx 45^\circ$ ).

**Additional constraints.** To ensure that the simplified mesh preserves the ridges and other critical features, we add two other constraints. We define a cylinder surrounding the critical curve (analogous to the envelope for a surface), its radius corresponding to the desired Hausdorff distance  $\delta$ . We also consider a local cone centered at the vertices to ensure the smoothness property defined above, its axis corresponding to the normal at the curve vertex.

**Local mesh modifications.** The extraction of a geometric surface mesh is based on local operations. As indicated, mainly edge collapsing and edge swapping operations are performed. The latter operation is applied only if the two triangles sharing a mesh edge are almost coplanar, so as to preserve the geometric approximation.

In addition, a node relocation operation can also be applied after the edge collapsing stage in order to improve the element shape quality. For a given vertex  $P$ , the procedure consists in moving  $P$  step by step towards an *optimal* point. Following the idea in [12], the geometry of the surface

around  $P$  is locally approached by a quadric surface and the optimal point is projected onto this surface (see Section 3.1 for more details on how to construct such a quadric surface). The same evaluation procedure as above can be used to control the geometric deviation during this stage.

**Simplification algorithm.** Schematically, the geometric simplification algorithm can be written as follows :

```

Initialisations
  set  $b_i(K_j^0) = 0, \forall K_j^0 \in M_0 = T,$ 
  set  $\delta_i$  to  $\delta_0$ 
while  $\delta_i < \delta$ 
  collapse edges of  $M^i$  if geometry
    and mesh quality are preserved,
  optimize mesh quality (edge swapping),
  optimize the element shape qualities
    using node relocation,
  increment  $\delta_i$ 
end while
global node smoothing.

```

### 2.3 Example.

We give now an example of this approach on a CAD model. In the example in Figure 3 (left), the initial triangulation has been obtained using an octree-based reconstruction algorithm [20].

**Remark 2.1** *Notice also that the simplification scheme induces an anisotropic metric related to the local principal radii of curvatures of the surface (as will be seen below).*

## 3. CURVATURES EVALUATION

In order to construct a *unit mesh*, we need to evaluate the intrinsic properties of the underlying surface. This involves computing the principal curvatures and principal directions at mesh vertices. Notice that the previous simplification procedure does not explicitly need this information to extract a geometric mesh (this interesting feature makes possible to process noisy data).

The principal curvatures at a point  $P$  of a  $C^2$  surface can be computed numerically based on the

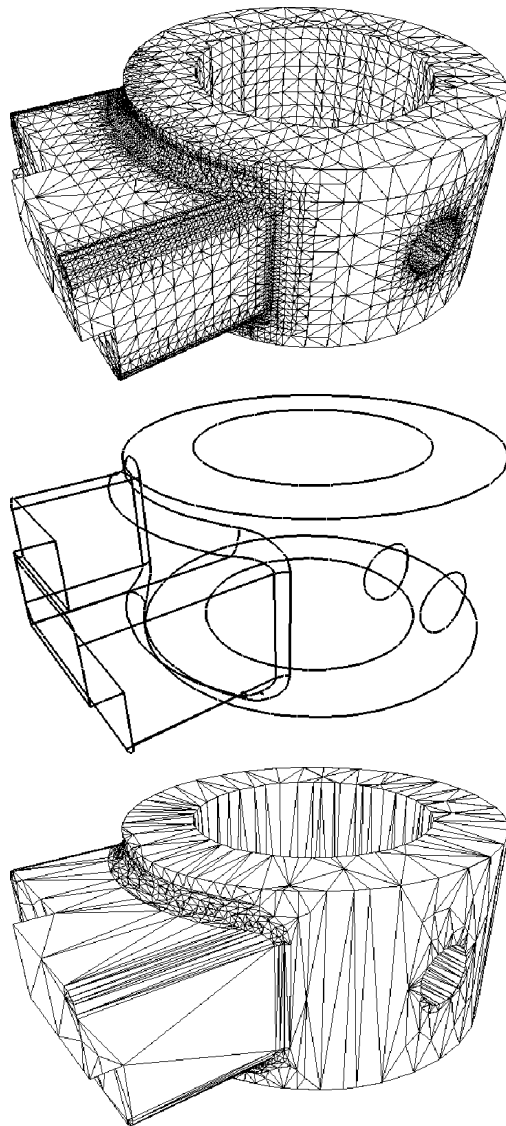


Figure 3. *Example of extraction of a geometric mesh. The initial triangulation (top, data courtesy of Mac Neal-Schwendler Corp.) has been processed to identify the geometric features (middle) before constructing a geometric mesh (bottom).*

given surface triangulation. To this end, the underlying surface geometry is locally approached by a quadric surface [12].

### 3.1 Construction of a quadric surface

Our approach is based on a least-squares fit of adjacent mesh vertices to compute this quadric surface. At each mesh vertex  $P$ , a vertex unit normal vector  $\vec{\nu}(P)$  is first initialized as a (weighted) average of the incident face normals. At  $P$ , the tangent vector plane  $\Pi(P)$  is directed by the two vectors  $(\vec{\tau}_1, \vec{\tau}_2)$  forming an orthonormal basis. The normal  $\vec{\tau}_1 \wedge \vec{\tau}_2$  to the plane coincides with the normal  $\vec{\nu}$  at  $P$ . So, the unit normal  $\vec{\nu}$  and the vectors  $\vec{\tau}_1$  and  $\vec{\tau}_2$  form a local system of coordinates :  $\mathcal{F} = (P, \vec{\tau}_1, \vec{\tau}_2, \vec{\nu})$ , the so-called *local frame* at  $P$ .

The quadric surface  $\Sigma$  of class  $c$  ( $c \geq 2$ ) centered at  $P$  is represented in the explicit form  $z = z(u, v)$ . Considering a surface point  $M = (x, y, z)$  and the point  $P = (0, 0, 0)$  as origin of the coordinate system (the  $z$ -axis is in the same direction as the surface normal) leads to write the equation :

$$F(x, y, z) = ax^2 + bxy + cy^2 - z = 0, \quad (7)$$

where  $a, b, c$  are three coefficients to be determined. To find this surface, we consider all vertices  $Q_i = (x_i, y_i, z_i)$  that belong to the ball  $\mathcal{B}(P)$  of  $P$  (*i.e.*, that are connected to  $P$  by an edge  $PQ_i$ ) and assume that the quadric surface must fit at best these points. We obtain a (usually over determined) linear system of  $m$  equations ( $m = \text{Card}(\mathcal{B}(P))$ ). Solving this system is equivalent of minimizing the following sum ( $a, b, c$  are solutions of an optimization problem) :

$$\min \sum_i^m (ax_i^2 + bx_iy_i + cy_i^2 - z_i)^2, \quad (8)$$

that corresponds to minimizing the square of the norm of the distances to the quadric surface. In matrix form, this system can be written as :

$$\begin{pmatrix} u_1^2 & u_1v_1 & v_1^2 \\ \vdots & \vdots & \vdots \\ u_m^2 & u_mv_m & v_m^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_m \end{pmatrix}, \quad (9)$$

where  $d_i$  represents the difference vector (distance from point  $Q_i$  to  $\Pi(P)$ ). Classically, the general formula for computing the least squares solution of this kind of system  $AX = B$  consists of writing

$${}^tAA\bar{X} = {}^tAB,$$

where  $\bar{X}$  denotes the least squares solution of  $X$  in the equation. The resulting normal equations are then solved using a classical Gauss method.

The quadric surface being locally defined at each mesh vertex  $P$ , it can be used to compute the local principal curvatures at the mesh vertices.

### 3.2 Principal curvatures estimation

In classical differential geometry [17], provided the surface is at least of order 2, the coefficients  $E, F, G$  and  $L, M, N$  of the first and second fundamental forms of the surface at a point  $P$  are such that :

$$\begin{aligned} \Phi_1(\vec{V}) &= Edu^2 + 2Fdudv + Gdv^2, \\ \Phi_2(\vec{V}) &= Ldu^2 + 2Mdudv + Ndv^2, \end{aligned} \quad (10)$$

where  $\vec{V} = \lambda\vec{\tau}_1 + \mu\vec{\tau}_2$  denotes a vector of  $\Pi(P)$ . Knowing the quadric coefficients  $a, b, c$ , it becomes easy to compute the principal coefficients :

$$\begin{aligned} E &= 1 + (2au + bv)^2, & L &= 2a, \\ F &= (2au + bv)(bu + 2cv), & M &= b, \\ G &= 1 + (bu + 2cv)^2, & N &= 2c. \end{aligned} \quad (11)$$

**Principal curvatures.** At point  $P = (0, 0)$  in the local frame, we have :

$$\begin{cases} E = 1, & F = 0, & G = 1, \\ L = 2a, & M = b, & N = 2c. \end{cases} \quad (12)$$

The analysis of the variations of the normal curvature function

$$\kappa_n(\vec{\tau}) = \frac{\Phi_1(\vec{\tau})}{\Phi_2(\vec{\tau})}$$

leads to resolve the following equation :

$$\det \begin{vmatrix} \mu^2 & -\lambda\mu & \lambda^2 \\ E & F & G \\ L & M & N \end{vmatrix} = 0. \quad (13)$$

This equation admits (in principle) two distinct solutions, 2 pairs  $(\lambda_1, \kappa_1), (\lambda_2, \kappa_2)$ . The extrema values  $\kappa_1$  and  $\kappa_2$  of  $\kappa_n$  (*i.e.*, the roots of the equation) are the so-called *principal curvatures* of the surface at  $P$ . Comparing Equation (13) with the second order equation :

$$\kappa^2 - (\kappa_1 + \kappa_2)\kappa + \kappa_1\kappa_2 = 0,$$

yields :

$$\begin{aligned} \kappa_1\kappa_2 &= \frac{LN - M^2}{EG - F^2} = 4ac - b^2, \\ \kappa_1 + \kappa_2 &= \frac{NE - 2MF + LG}{EG - F^2} = 2(a + c), \end{aligned} \quad (14)$$

where  $K = \kappa_1 \kappa_2$  is called the *Gaussian curvature* and  $H = \frac{1}{2}(\kappa_1 + \kappa_2)$  is called the *mean curvature* of the surface at  $P$ . Replacing  $E, F, G, L, M, N$  (from Equation (11)) allows to find the extrema values  $\kappa_1$  and  $\kappa_2$  at  $P$  :

$$\kappa_i = \frac{2(a+c) \pm \sqrt{\Delta}}{2}, \quad (15)$$

with  $\Delta = (2(a+c))^2 - 4(4ac - b^2)$ .

**Principal directions.** Equation (13) admits two distinct solutions (in the general case), that define two directions in the  $(u, v)$  plane. The corresponding directions in the tangent plane are the *principal directions*. Hence, solving Equation 13 leads to :

$$\lambda_{1,2} = \frac{2(c-a) \pm \sqrt{\Delta'}}{2b}, \quad (16)$$

with  $\Delta' = 4((a-c)^2 + b^2)$ . To find the expressions of the two vectors, using Euler's theorem and considering the relation :

$$\lambda_{1,2} = \frac{du}{dv} = \tan(\theta),$$

in the local frame leads to find the two desired vectors in the global frame :

$$\begin{cases} \vec{\lambda}_1 &= e_{1,x} \cdot \vec{\tau}_1 + e_{1,y} \cdot \vec{\tau}_2, \\ \vec{\lambda}_2 &= e_{2,x} \cdot \vec{\tau}_1 + e_{2,y} \cdot \vec{\tau}_2, \end{cases} \quad (17)$$

with

$$\vec{e}_1 = {}^t(\cos(\theta), \sin(\theta)), \quad \vec{e}_2 = {}^t(-\sin(\theta), \cos(\theta)).$$

**Local behavior of the surface.** The principal curvatures at point  $P$  can be used to characterize the local behavior of the surface :

- if  $\kappa_1 \kappa_2 > 0$ , the point  $P$  is *elliptic*,
- if  $\kappa_1 \kappa_2 < 0$ , the point  $P$  is *hyperbolic*,
- if  $\kappa_1 \kappa_2 = 0$ , the point  $P$  is *parabolic*.

This information can be used to govern the local mesh modification operators (*i.e.*, to preserve the point characteristic).

### 3.3 Definition of a geometric metric $\mathcal{G}_3$

A so-called *geometric metric*, denoted as  $\mathcal{G}_3$ , is constructed in the tangent planes of the vertices of the geometric surface mesh. This *discrete metric*<sup>2</sup> allows to bound the gap between any mesh edge and the surface by a given threshold value  $\varepsilon$ . A matrix of the form :

$$\mathcal{G}_3(P)_{\rho_1, \rho_2} = {}^t \mathcal{D}(P) \begin{pmatrix} \frac{1}{\alpha^2 \rho_1^2(P)} & 0 & 0 \\ 0 & \frac{1}{\beta^2 \rho_2^2(P)} & 0 \\ 0 & 0 & \lambda \end{pmatrix} \mathcal{D}(P), \quad (18)$$

where  $\mathcal{D}(P)$  corresponds to the principal directions at  $P$ ,  $\rho_1 = 1/\kappa_1$ ,  $\rho_2 = 1/\kappa_2$  are the main radii of curvature,  $\alpha$  and  $\beta$  are appropriate coefficients and  $\lambda \in \mathbb{R}$ , provides an *anisotropic* (curvature-based) control of the geometry.

As can be seen, this metric prescribes mesh sizes as well as element stretching directions at mesh vertices. In other words, the local size is proportional to the principal radii of curvature, the coefficient of proportionality being related to the largest allowable deviation gap between the mesh elements and the surface geometry [6]. For instance setting a constant gap values comes down to fixing :

$$\alpha = 2 \sqrt{\varepsilon(2 - \varepsilon)},$$

and to defining :

$$\beta = 2 \sqrt{\varepsilon \frac{\rho_1}{\rho_2} \left( 2 - \varepsilon \frac{\rho_1}{\rho_2} \right)}.$$

As the size may change rapidly from one vertex to another, the mesh gradation may not be bounded locally, thus leading to potential poorly-shaped elements. To overcome this problem, the size map  $\mathcal{G}_3$  is modified to a metric  $\tilde{\mathcal{G}}_3$  using a *size-correction* procedure [2].

**Remark 3..1** *Using an interpolation scheme over the (discrete) size map  $\mathcal{G}_3$ , it is possible to deduce the size at any point on the surface.*

In finite element computations, it is frequent to define a metric that allows an homogeneous distribution of the related related to the interpolation of the solutions. This computational metric  $\mathcal{M}_3$  can be then “intersected” with the metric  $\mathcal{G}_3$  so as to govern the unit mesh creation.

<sup>2</sup>as it is only defined at the mesh vertices.

## 4. THE GEOMETRIC SUPPORT

Once a geometric mesh has been obtained using the previous approach, a geometric support can be constructed internally. It represents the analytical definition of a surface and can be used to emulate the features of a geometric modeling system.

### 4.1 Construction of the support

The construction of the support involves the definition of a piecewise linear surface of order  $G^1$  based on the geometric mesh previously extracted. Each triangle represents a patch, two adjacent patches sharing a common tangent plane to ensure the desired continuity property (except if the common edge is a ridge).

Walton's method [22] appears appropriate to define the geometric support. Schematically, it consists in defining a network of patch boundary curves and their transverse tangent planes. To this end, the normals to the surface at the vertices are simply interpolated. A nice feature of this technique is that each patch can be defined independently, from its boundary, by taking the tangent planes into account and by using Gregory's approach [10]. Each curve and the related tangent planes are completely defined from the normals at its endpoints. More precisely, the transverse tangent planes are generated from the vectors at any point along these curves and from vectors resulting from a quadric interpolation of the binormal at the endpoints. Hence, each triangle is associated with a patch defined using a rational function in its interior and a quartic along its edges (Figure 4).

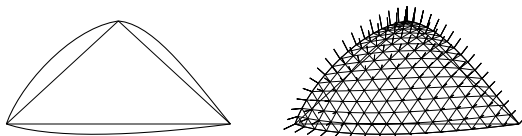


Figure 4. *Geometric support construction. Initial triangle with boundary curves (left-hand side) and geometric support based on Walton's patch (right-hand side).*

**Remark 4.1** *As the geometric mesh is an accurate piecewise linear approximation of the surface, the geometric support can be used to emulate a modeling system.*

### 4.2 Use of the support

Given a mesh vertex, the geometric support is used to supply the location of the closest point onto the surface from the point. Moreover, at a  $G^1$  continuous point, the surface normal and the principal radii of curvature can be returned by the support. For a point located along a ridge (where the tangent planes are not  $G^1$  continuous), the tangent to the curve at the point is returned.

This feature will be used during edge splitting operations.

## 5. UNIT SURFACE MESH CONSTRUCTION

This stage involves the construction of a unit mesh (with respect to a metric) via local geometrical and topological mesh modifications. We will now briefly describe this stage (for more details, the reader is referred to [7], for instance).

### 5.1 Length of a segment

**Anisotropic case.** In the general case, the desired element size and stretching directions at a vertex  $P$  is represented by a  $3 \times 3$  matrix  $\mathcal{M}_3$  as defined in Section 3.3. The restriction of a parametrized arc  $\gamma(t)$ ,  $t \in [a, b]$  to a vector  $\overrightarrow{AB}$  (with  $\gamma(t) = A + t\overrightarrow{AB}$ ) allows us to write the length  $l_{AB}$  of a segment as :

$$l_{AB} = \int_0^1 \sqrt{{}^t\gamma'(t) \mathcal{M}_3(A + t\overrightarrow{AB}) \gamma'(t)} dt \quad (19)$$

Noticing that  $\gamma'(t) = \overrightarrow{AB}$  leads to writing the relation :

$$l_{AB} = \int_0^1 \sqrt{{}^t\overrightarrow{AB} \mathcal{M}_3(A + t\overrightarrow{AB}) \overrightarrow{AB}}. \quad (20)$$

Considering that the metric  $\mathcal{M}_3$  is independent of the position makes it possible to reduce the problem to the Euclidean case and thus to write :

$$l_{AB} = \|\overrightarrow{AB}\|_{\mathcal{M}_3} = \sqrt{{}^t\overrightarrow{AB} \mathcal{M}_3 \overrightarrow{AB}}. \quad (21)$$

**Isotropic case.** In this case, the size at a vertex is constant in any direction. Hence, the metric at

a vertex  $P$  can be expressed as :

$$\mathcal{M}_3(P) = \frac{Id}{h^2(P)},$$

where  $h(P)$  is a scalar value representing the size at  $P$ . Considering a monotonuous size interpolation function  $H(t)$  such that  $H(0) = h(A)$  and  $H(1) = h(B)$ , the metric  $\mathcal{M}_3$  can be defined as :

$$\mathcal{M}_3(A + t\overrightarrow{AB}) = \frac{Id}{H(t)},$$

and then Equation (21) yields :

$$l_{AB} = \|\overrightarrow{AB}\| \int_0^1 \frac{1}{H(t)} dt. \quad (22)$$

This equation can be computed exactly for a given function  $H(t)$  (for instance corresponding to a geometric size progression along the edge).

## 5.2 Unit mesh

The metric  $\mathcal{G}_3$  is used to characterize the optimal length of any edge  $PX$  incident to  $P$ .

**Definition 5..1** A mesh  $M$  is said to conform the metric  $\mathcal{G}_3$  iif :

$$\frac{1}{\sqrt{2}} \leq l_{AB} \leq \sqrt{2} \quad \forall AB \in M. \quad (23)$$

Such a mesh is called a *unit mesh*.

To facilitate the evaluation of the resulting mesh, we introduce the *efficiency index*, defined as [8]:

$$\tau = 1 - \frac{\sum_{i=1}^{na} e_i^2}{na}, \quad (24)$$

where  $e_i = 1 - l_i$  if  $l_i < 1$  and  $e_i = 1 - \frac{1}{l_i}$  if  $l_i > 1$ ,  $na$  being the number of mesh edges. This coefficient seems adequate for a quick estimation of the size quality of the mesh with respect to a given size map. In practice, a mesh is considered as conforming the size map if  $\tau > 0.91$ .

## 5.3 Unit surface remeshing

The purpose of the surface optimization stage is to modify the geometric mesh so as to create a

unit mesh conforming a given metric  $\mathcal{M}_3$ . The latter can be either the geometric metric  $\mathcal{G}_3$  or any combination of  $\mathcal{G}_3$  and a user-supplied metric  $\mathcal{M}'_3$  (for instance provided by an error estimate in a finite element adaptive scheme). usually, the unit mesh must contain well-shaped elements as it will be used for numerical computations.

**Local mesh modifications.** The optimization of the mesh is based on local modification operators allowing to simplify (vertex or edge collapsing,...), to enrich (edge splitting) or to improve (node relocation, edge swapping,...). Usually, a local mesh modification is performed provided the geometric approximation is preserved and the mesh quality is not degraded.

Among these operators, those related to mesh enrichment or node smoothing require the knowledge of the underlying geometry (*i.e.*, via the geometric support), whereas others do not need this type of information [7].

Practically, the optimization procedure consists in analyzing the current mesh edges in order to collapse the short edges and to split the large ones. An edge is considered as short (resp. long) if its length is lesser (resp. larger) than  $1/\sqrt{2}$  (resp.  $\sqrt{2}$ ). This procedure is based on edge splitting, edge collapsing and edge swapping operations. Moreover, several geometric measures are introduced to control the deviation between the mesh elements and the surface geometry as well as the element shape quality [6].

More precisely, the edge collapsing operation consists in identifying the two endpoints of an edge into a single vertex. This operation removes the two triangles sharing the edge and constructs new triangles incident to the remaining vertex. The final configuration is known *a priori* and can be validated before effectively applying the operator.

The edge splitting operation consists in introducing a vertex along a mesh edge and snapping this point onto the surface so that, at least, one subsegment resulting from this operation is of unit length. The two triangles sharing the edge are replaced by four <sup>3</sup> new triangles. Local edge swappings are then applied on the neighboring edges to improve the mesh quality.

---

<sup>3</sup>or more than four if the edge is non-manifold.

The point relocation procedure consists in redefining all triangles sharing a vertex  $P$ . The optimal position of this vertex is determined by avering the optimal points for the triangles in the ball of  $P$  (the points leading to create equilateral triangles). This provides a point  $P'$  towards which the point  $P$  is moved. Instead of moving  $P$  in the tangent plane, we use the paraboloid defined in Section 3.1 as the local approximation of the surface (*i.e.*, a second order approximation of the surface).

Edge swapping is performed only if the two triangles sharing the edge are almost coplanar. Actually, this operation leads to locally change the surface curvature along the edge and thus may be quite troublesome for further processing.

**General scheme** The optimization procedure modifies iteratively the current mesh, in order to adapt the element sizes to the prescribed size map. The main steps of this approach can be summarized as follows :

```

Preliminary definitions :
  construction of the geometric support,
  definition of the metric  $\mathcal{G}_3$ ,
  control of the mesh gradation,
while  $M^i$  is modified
  compute  $l_{AB}$ ,
  if  $l_{AB} < \frac{1}{\sqrt{2}}$ ,
    collapse  $AB$ ,
  else if  $l_{AB} > \sqrt{2}$ ,
    split  $AB$ ,
  endif
end while
shape quality optimization
  (node relocation, edge swapping.)

```

## 6. APPLICATION EXAMPLES

In this section, application examples are provided to illustrate this approach of unit surface remeshing. To help appreciate the results, statistics tables are provided for each example :  $np$  and  $ne$  denote the number of mesh vertices and triangles;  $Q_{max}$ ,  $Q_{avg}$  indicate respectively the worst and average element quality and  $Q_{[12]}$  indicate the percentage of element having a quality better than 2. Less than 20 seconds (including i/o, HP

782, 180Mhz) have been necessary to complete the meshes.

- *Creation of a finite element mesh.*

Figure 5 illustrates the two stages involved in the creation of a finite element mesh of a mechanical device. At first, a geometric mesh (middle) is created from a given surface triangulation (init, top), corresponding to a tolerance  $\delta = 1/1000$  of the bounding box size (top). Then, a geometric support is defined using this geometry. Finally, based on the geometric metric  $\mathcal{G}_3$  (isotropic in this case), a finite element mesh is constructed (F.E., bottom).

mesh	$np$	$ne$	$Q_{max}$	$Q_{avg}$	$Q_{[12]}$
init.	1,685	3,410	5.82	1.43	90.5
geom.	2,798	5,636	10.5	1.7	81.1
F.E.	4,748	9,536	9.8	1.4	94.

Table 1. Statistics for the creation of a finite element mesh.

- *Control of the element shape quality.*

For numerical computations, the shape quality of the mesh elements is usually important. Therefore, to avoid rapid size variations, a correction procedure is applied to the geometric size map. On Figure 6, left-hand side, the mesh has been optimized with respect to the geometric size map  $\mathcal{G}_2$ . On the right-hand side however, the size map was modified to control and bound the mesh gradation to a value of 1.5, for the same geometric deviation ( $\varepsilon = 11^\circ$ ).

mesh	$np$	$ne$	$Q_{avg}$	$Q_{[12]}$
no $H_c$	20,445	40,942	1.62	85.5 %
$H_c$	35,827	71,706	1.3	97. %

Table 2. Statistics for the control of the mesh gradation.

- *Mesh simplification.*

A nice application of this approach concern mesh decimation which aims at finding a good compromise between the number of mesh elements

and the accuracy of the geometric approximation. Here, mesh simplification is considered as a specific case of unit remeshing. The geometric metric is simply modified to correspond to a larger deviation value.

Figure 7 illustrates a biomedical application. The initial surface triangulation is a polyhedral reconstruction from MRI data (data courtesy Low Temp. Lab., Helsinki Univ. Technology, Sweden). The objective is to create a geometric mesh suitable for biomedical simulation of inverse problems in MEG. The original mesh contains 67,106 vertices and 134,212 triangles and is way to large to be numerically exploitable.

Figure 8 illustrates an interesting and promising application of mesh simplification: the construction of hierarchies of triangle meshes for visualization purposes.

- *Mesh enrichment.*

Contrary to mesh decimation, it may be desirable in some applications to enrich the given triangulation. Figures 9 and 10 illustrate this issue on a  $C^1$  continuous biomedical example and on a mechanical model having ridges and constrained entities. This example shows the role of the geometric support in point insertion.

- *Mesh adaptation.*

The convergence, the accuracy and the error estimate theorems for finite element computations are all related to the parameter  $h$  representing the mesh element size. In mesh adaptation, the current mesh is locally modified so as to conform a prescribed size map corresponding to a combination between the geometric size map and a size map provided by an *a posteriori* error estimate.

Figure 11 illustrates three stages of mesh adaptation based on an analytical field.

## 7. CONCLUSIONS

In this paper, we have proposed a general scheme for constructing geometric surface meshes from a given arbitrary surface triangulation.

To this end, the geometry is first extracted based on the evaluation of the Hausdorff distance from

the initial triangulation. Then, a  $G^1$  continuous support is constructed and a geometric size map is defined based on the geometric properties of the mesh. Finally, geometric finite element surface meshes can be constructed with respect to the size map. This approach proved to be useful for dealing with smooth surfaces as well as with CAD models.

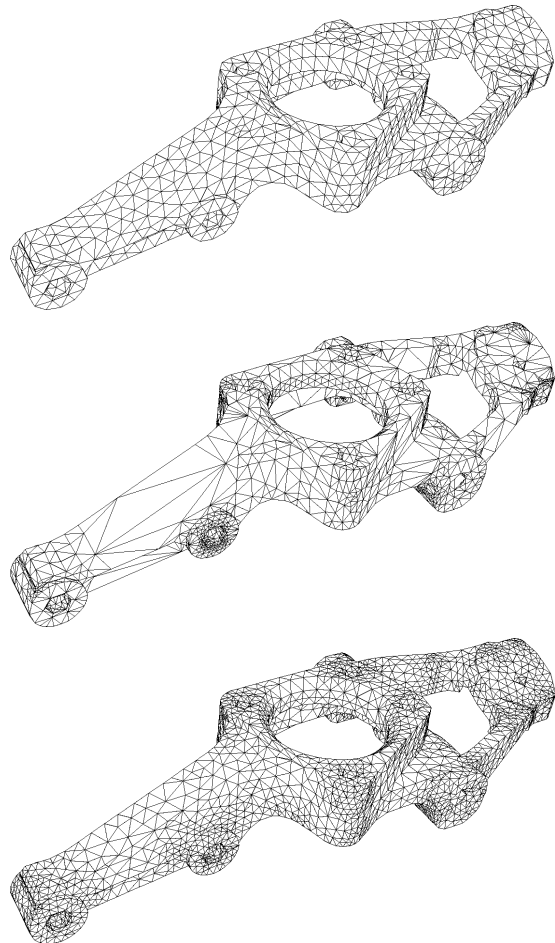


Figure 5. *Three stages in the construction of a finite element mesh. Initial surface triangulation (top, data courtesy of Mac Neal-Schwendler Corp.), geometric mesh (middle) and resulting computational mesh (bottom).*

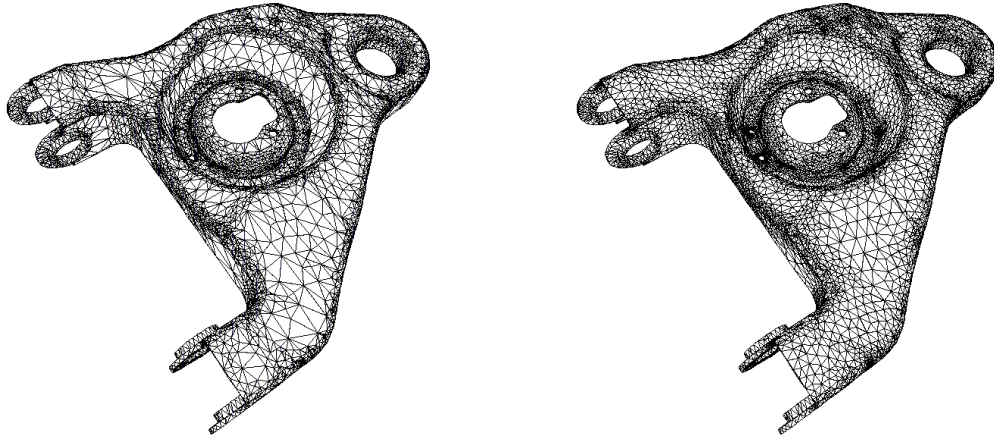


Figure 6. *Influence of the metric correction procedure. Geometric surface mesh without gradation control (top) and with gradation control  $H_c = 1.2$  (bottom).*

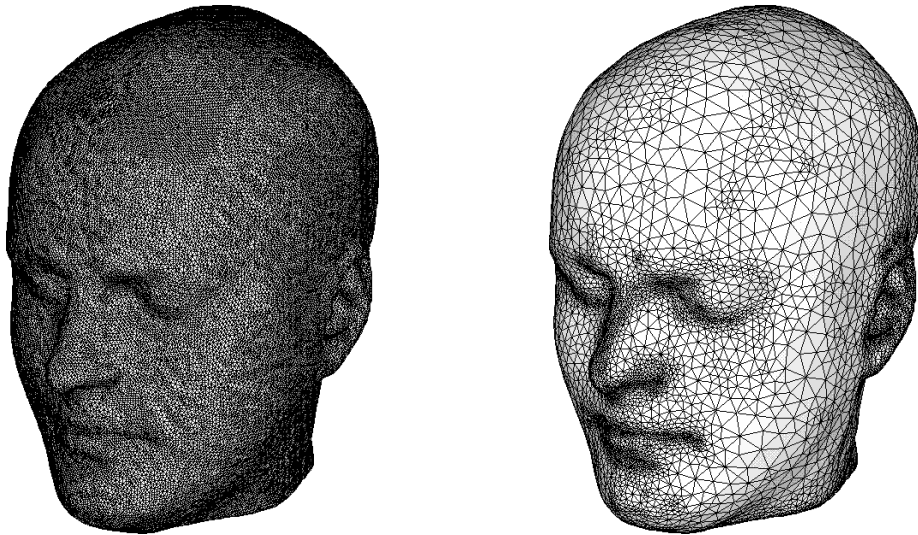


Figure 7. *Example of mesh decimation : initial triangulation (top) and geometric finite element mesh (bottom).*

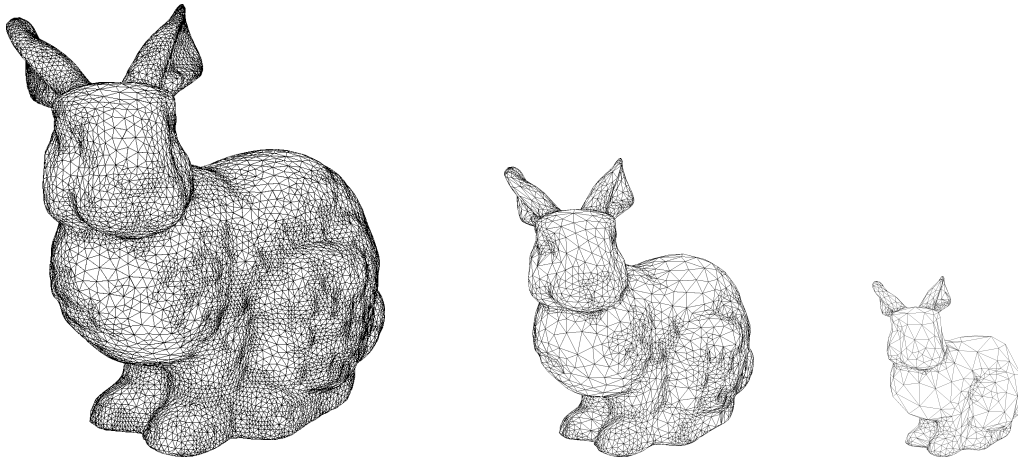


Figure 8. *Example of mesh decimation in context of multiresolution hierarchies.*

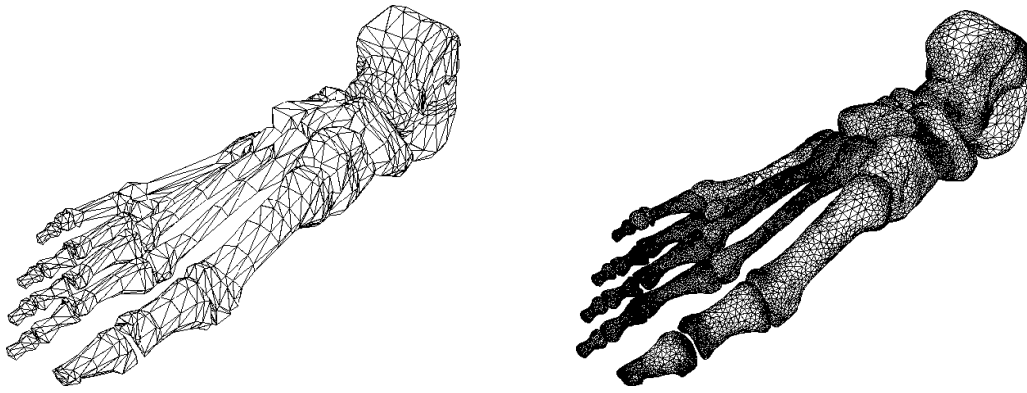


Figure 9. *Example of mesh enrichment for a smooth surface.*

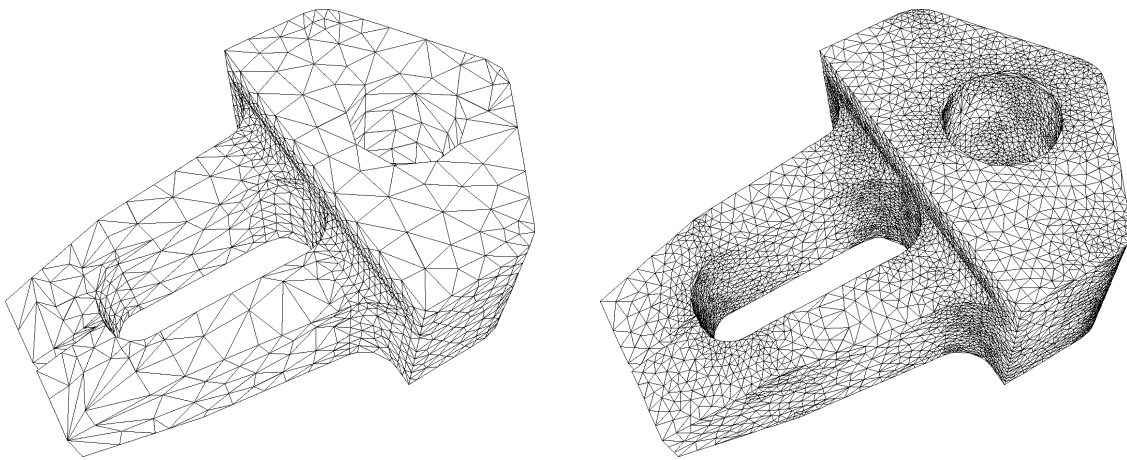


Figure 10. *Example of mesh enrichment for a mechanical device.*

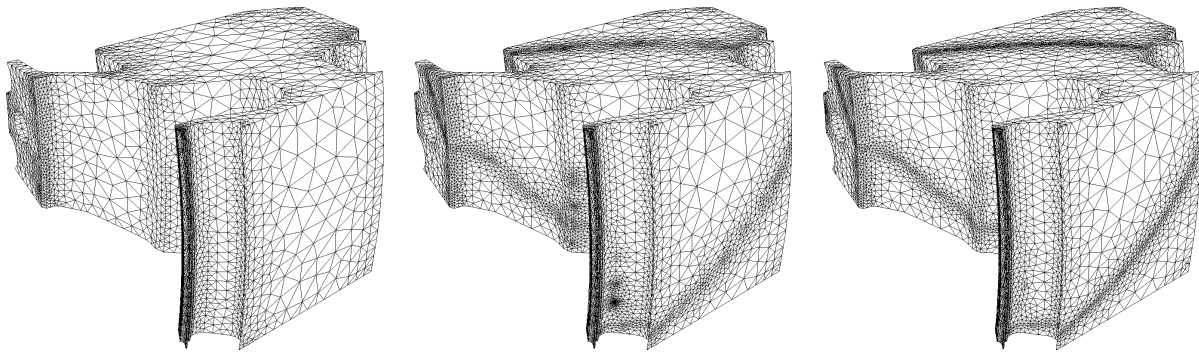


Figure 11. *Example of surface (left-hand side) and volume (right-hand side) mesh adaptation (3 iterations have been necessary).*

## REFERENCES

- [1] H. BOROUCAKI (1999), Simplification de maillages basée sur la distance de Hausdorff, *C.R. Acad. Sci.*, t. 329, Série I, 641-646.
- [2] H. BOROUCAKI, F. HECHT AND P.J. FREY (1998), Mesh gradation control, *Int. J. Numer. Methods Eng.*, **43**(6), 1143-1157.
- [3] H. BOROUCAKI AND P.L. GEORGE (2000), Quality mesh generation, *C.R. Acad. Sci. Paris*, **328**, Série IIb, 505-518.
- [4] P.G. CIARLET (1991), Basic Error Estimates for Elliptic Problems, in Handbook of Numerical Analysis, vol II, Finite Element methods (Part 1), P.G. Ciarlet and J.L. Lions Eds, North Holland, 17-352.
- [5] J. COHEN, A. VARSHNEY, D. MANOCHA, G. TURK, H. WEBER, P. AGARWAL, F. BROOKS AND W. WRIGHT (1996), Simplification envelopes, *Comput. Graphics*, Proc. Siggraph'96, 119-128.
- [6] P.J. FREY AND H. BOROUCAKI (1998), Surface mesh quality evaluation, *Int. J. Numer. Methods Engng*, **45**, 101-118.
- [7] P.J. FREY AND H. BOROUCAKI (1998), Geometric surface mesh optimization, *Computing and Visualization in Science*, **1**, 113-121.
- [8] P.J. FREY ET P.L. GEORGE (1999), Maillages : applications aux éléments finis, Hermes Science Publications, Paris (also in english, Mesh Generation, Hermes Science Publications, may 2000).
- [9] M. GARLAND AND P.S. HECKBERT (1997), Surface simplification using quadric error metrics, *Siggraph '97*, 209-218.
- [10] J.A. GREGORY (1974), Smooth interpolation without twist constraints, in R.E. Barnhill and R.F. Riesenfeld (Eds.), *Computer Aided Geometric Design*, Academic Press, 71-87.
- [11] A. GUÉZIEC (1996), Surface simplification inside a tolerance volume, *IBM Research Report*, RC-20440.
- [12] B. HAMANN (1993), Curvature approximation for triangulated surfaces, in *Geometric Modelling*, Computing Suppl. 8, Farin, Hagen, Noltmeier and Knodel eds., Springer, NY, 139-153.
- [13] B. HAMANN (1994), A data reduction scheme for triangulated surfaces, *Comp. Aided Geom. Des.*, **11**, 197-214.
- [14] H. HOPPE, T. DE ROSE, T. DUCHAMP, J. McDONALD AND W. STUETLZE (1993), Mesh optimization, *Computer Graphics*, Proc. Siggraph'93, 19-26.
- [15] H. HOPPE (1999), New quadric metric for simplifying meshes with appearance attributes, *IEEE Visualization 1999*, Oct. 1999, 59-66.
- [16] R. KLEIN, G. LIEBICH AND W. STRASSER (1996), Mesh reduction with error control, *Proc. IEEE Visualization'96*, 311-318.
- [17] J. LELONG-FERRAND AND J.M. ARNAUDIÈS (1977), *Cours de Mathématiques*, Tome 3, *Géométrie et cinématique*, Dunod Université, Bordas.
- [18] K. REINHARD (1998), Multiresolution representations for surfaces meshes based on the vertex decimation method, *Comput. Graphics*, **22**(1), 13-26.
- [19] W.J. SCHROEDER, J.A. ZARGE AND W.E. LORENSEN (1992), Decimation of triangle meshes, *Computer Graphics (Proc. Siggraph)*, **26** (2), 65-70.
- [20] M.S. SHEPHARD AND M.K. GEORGES (1991), Automatic three-dimensional Mesh Generation by the Finite Octree Technique, *Int. j. numer. meth. eng.*, **32**, 709-749.
- [21] G. TURK (1992), Re-tiling polygonal surfaces, *Computer Graphics*, **26**, (2), 55-64.
- [22] D.J. WALTON AND D.S. MEEK (1996), A triangular  $G^1$  patch from boundary curves, *Comp. Aided Design*, **28**, 113-123.