



Simplification of surface mesh using Hausdorff envelope

H. Borouchaki ^{a,c,*}, P.J. Frey ^{b,c}

^a *University of Technology of Troyes, GSM-LASMIS, BP 2060, 10010 Troyes Cedex, France*

^b *Université Pierre et Marie Curie, Laboratoire Jacques Louis Lions, 75252 Paris Cedex 05, France*

^c *INRIA Rocquencourt, Gamma Project, BP 105, 78153 Le Chesnay Cedex, France*

Received 8 December 2003; received in revised form 24 June 2004; accepted 26 November 2004

Abstract

This paper presents a new method for surface mesh simplification. Given an initial surface mesh (denoted as reference mesh), the goal is to reduce the number of mesh elements while preserving the geometric approximation as well as the shape quality of the resulting mesh. Two tolerance areas with respect to the reference mesh have been introduced to preserve the geometry of the surface. The reference mesh is then simplified and optimized (in terms of shape quality) so that the resulting mesh belongs to these tolerance areas. Several examples of surface meshes are provided regarding different application areas in order to assess the efficiency of the simplification method.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Mesh simplification; Decimation; Surface meshing; Hausdorff distance

1. Introduction

Surfaces can be defined essentially in two different ways for geometric modeling of objects depending on the application areas envisaged: computer graphics, finite element computations, etc. The first way, mostly used in CAD modelers, consists in defining a set of conformal parametric patches, a complex object then requires a large number of such patches to be accurately described. In most cases, a patch dependent algorithm will preserve the contours of each patch. For finite element computation however, it is often desirable to eliminate redundant information as well as small “geometric features” of the object, either from the CAD model or directly from the mesh. The second way consists in defining the surface via a triangulation,

* Corresponding author. Address: University of Technology of Troyes, GSM-LASMIS, BP 2060, 10010 Troyes Cedex, France. Tel.: +33 25 71 56 67; fax: +33 25 71 56 75.

E-mail address: Houman.Borouchaki@utt.fr (H. Borouchaki).

i.e., a mesh created using a reconstruction method based on a set of sampled surface points (e.g. provided by sensing devices) or on volumetric data (e.g. provided by scanning devices). Obviously, the accurate definition of a complex object requires a high-accuracy data acquisition, thus leading to a (unnecessary) large number of mesh elements. Typically modern data acquisition techniques lead to meshes containing of about several millions of elements (e.g. the digital Michelangelo project, Stanford University [1] and the Visual Human Project, National Library of Medicine).

1.1. Related work

As stated above, for graphic and numerical simulations, it is usually desirable to significantly reduce the complexity of the discrete surface representation (i.e., the mesh size) while preserving, as much as possible, the geometry of the object. In other words, the challenge is to eliminate the geometric redundancies while not altering the geometric properties of the model. Modern mesh simplification methods attempt to answer these two requirements. Actually, they are mostly based on the optimization of geometric criteria [2,5,7,9,10,12,14,15]. The applications potentially concerned range from computer graphic (where it is important to deal with the notion of budget polygon, i.e. a bound on the number of polygons that can be interactively rendered on graphics workstations) to numerical simulations (e.g. finite element computations), including scientific visualisation, data compression, etc.

A surface mesh (where element vertices are considered to belong to the surface) is considered as geometrically suitable if (i) all mesh elements are close to the surface and (ii) each mesh element is close to the tangent plane of the surface at its vertices [4]. The first (proximity) property allows to bound the gap between the elements and the surface. This gap measures the largest distance between any point of an element and the surface. The second (smoothness) property ensures that the surface is locally G^1 continuous. Hence, the angular deviation between the element and the tangent plane at its vertices shall be bounded. An optimal surface mesh is such that the elements are close to regular or equilateral (which is an essential requisite for most numerical applications).

1.2. General scheme

In this paper, we present a new mesh simplification method based on the discrete evaluation of the Hausdorff distance between two meshes, notably inspired by the ideas developed in [3]. Schematically, it consists in three stages. At first, a global tolerance envelope is defined around the surface as well as a local tolerance cone centered at each vertex of the reference surface mesh. Then, the edges of the initial surface mesh are iteratively analyzed and eventually removed if the resulting elements do not violate the tolerance requirements. Actually, the tolerance areas are introduced to enforce the proximity and the smoothness properties introduced hereabove. Finally, after each vertex or edge removal operation, the current mesh quality is optimized with respect to the element shape measure using edge flipping and point relocation procedures, provided the geometric accuracy is preserved.

1.3. Paper outline

Section 2, the definition of the local tolerance areas is introduced as well as an approximation of the Hausdorff distance used to control the simplification process. Section 3, the overall scheme of the method is described in the context of smooth surfaces, and an extension of the algorithm for other (i.e., non smooth) surfaces is presented. Section 4, a method to numerically measure the distance between two given meshes is described, that can be used to validate the simplification method. Finally, Section 5, several application examples are introduced to emphasize the efficiency of the simplification algorithm.

2. Definition of the tolerance areas

As explained in the previous section, the control of the simplification process relies on a local tolerance property. To enforce this proximity criterion, a global proximity region is defined around the surface at a given (Hausdorff) distance δ on both sides of the initial reference surface triangulation. This distance can be expressed in percentage of the minimal bounding box size of the initial mesh. Let Σ denotes the initial surface. Then, the surface $\Sigma(\delta)$ located at a Hausdorff distance δ of Σ is the geometric envelope of the spheres $\mathcal{S}(P, \delta)$ centered at points P and of radius δ (P covering the set of points of Σ , cf. Fig. 1).

Similarly, to enforce the smoothness (regularity) property, a local cone is centered at each vertex of the reference mesh, the normal to the surface at this vertex defines the principal axis of this cone, the aperture angle θ being given (cf. Fig. 2).

Each triangle resulting from mesh modification must, on the one hand, belong to the proximity envelope and, on the other hand, have a normal that is contained within the regularity cones associated with the mesh vertices (this last property ensuring that the triangle is close to the tangent planes at each of its three vertices).

We briefly recall the mathematical formulation of the Hausdorff distance. The distance from a point X in \mathbb{R}^3 to a closed bounded set F in \mathbb{R}^3 is expressed by the relation:

$$d(X, F) = \inf_{Y \in F} d(X, Y),$$

where $d(.,.)$ denotes the usual Euclidean distance. Let F_1 and F_2 be two closed bounded sets in \mathbb{R}^3 and let denote $\rho(F_1, F_2)$ the quantity

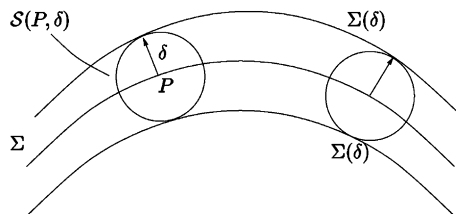


Fig. 1. Surfaces located at a Hausdorff distance δ of the surface Σ .

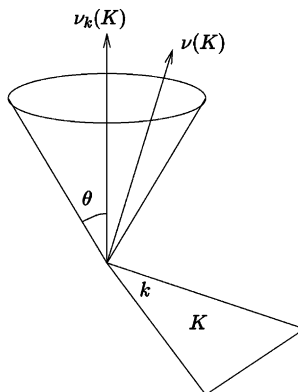


Fig. 2. Local cone with an aperture angle θ defined at a mesh vertex.

$$\rho(F_1, F_2) = \sup_{X \in F_1} d(X, F_2),$$

then, the Hausdorff distance $d_H(F_1, F_2)$ between F_1 and F_2 is defined as:

$$d_H(F_1, F_2) = \sup(\rho(F_1, F_2), \rho(F_2, F_1)).$$

Let also $\mathcal{T}^{\text{ref}} = \bigcup_i K_i^{\text{ref}}$ denotes the reference mesh where K_i^{ref} represents a mesh triangle. Any triangle K resulting from the modification of \mathcal{T}^{ref} must comply with the two following relations:

$$d_H(K, \mathcal{T}^{\text{ref}}) \leq \delta \quad \text{and} \quad \forall k \langle v_k(K), v(K) \rangle \geq \cos \theta,$$

where $v(K)$ denotes the unit normal to triangle K and $v_k(K)$ denotes the unit normal to the surface at the k th vertex of K ($\langle \cdot, \cdot \rangle$ is the standard scalar product in \mathbb{R}^3), cf. Fig. 2.

We shall notice that the exact computation of the Hausdorff distance $d_H(K, \mathcal{T}^{\text{ref}})$ between a triangle K and the initial reference mesh is extremely costly. In the following section, we will show how to get a cheaper approximation (actually an upper bound) of this quantity, in special configurations of mesh optimization, by accounting for the distances in a hierarchical manner.

To summarize, the two tolerance regions introduced previously allow to control the gap (deviation) between the initial mesh and the simplified mesh of the surface as well as to ensure the geometric regularity (smoothness) of the resulting mesh.

3. Overview of the method

Here we assume (without any lack of generality) that the initial reference mesh is only defined by the list of its vertices and the list of its elements (triangles). The proposed method mainly consists in iteratively removing and optimizing all mesh edges, until no modification can occur. To this end, two local mesh modification operations are involved: the vertex identification (the two endpoints of an edge are merged together) and the edge flip.

The first procedure, while simple to describe and implement, is rather efficient to carry out on mesh edges, as the two properties (i.e., proximity and regularity) can be explicitly checked a priori (i.e., without actually doing the modification on the mesh structure). At each iteration, a mesh edge is removed and a new mesh (known a priori) is locally constructed. The operation is carried out if, on the one hand, the new mesh preserves the geometry of the surface and, on the other hand, if the shape quality of the mesh elements is not degraded too much. When a mesh edge has been removed, the second procedure (edge flip) is checked against each newly created mesh edge and eventually applied if the mesh quality is improved. This edge processing is usually followed by a node relocation procedure to optimize the element shape quality.

All these procedures will now be described in the case of smooth (sufficient geometric continuity order) surfaces. In particular, the general scheme of the method is explained and the extension to the case of surfaces presenting geometric discontinuities is discussed.

3.1. Smooth surface simplification

A “smooth” surface is a surface such that the tangent plane at each point is continuous. Typically, at each point of the surface, a unique normal vector is defined, that characterizes the tangent plane to the surface at this point. Let first consider an initial reference mesh \mathcal{T}^{ref} of such a surface. We suggest to simplify \mathcal{T}^{ref} using the iterative edge removal operation, as much as possible.

3.1.1. Edge removal

Let $\mathcal{T} = \bigcup_i K_i$ denotes the resulting mesh after at a given iteration, K_i representing the mesh elements (at iteration $j = 0$, $\mathcal{T} = \mathcal{T}^{\text{ref}}$). Let PQ represents an edge in the mesh \mathcal{T} and let $\mathcal{B}(P) = \bigcup_{K \ni P} \{K\}$ defines the “ball” of vertex P (i.e., the set of triangles connected to P) at a given iteration. The identification of P to Q (that leads to the removal of edge PQ) consists in replacing the triangles $\bigcup_{K \ni P} \{K\}$ connected to P by a new set of triangles $\bigcup_{K' \ni Q} \{K'\}$ defining the region $\mathcal{R}(Q)$. By doing the identification of P with Q , the geometry is preserved if each newly created triangle $K' \in \mathcal{R}(Q)$ complies with, on the one hand:

$$d_H(K', \mathcal{T}^{\text{ref}}) \leq \delta$$

and, on the other hand:

$$\langle v_k(K'), v(K') \rangle \geq \cos \theta,$$

where θ corresponds to a given tolerance angle.

Moreover, the following inequality can be established:

$$d_H(K', \mathcal{T}^{\text{ref}}) \leq d_H(K', \mathcal{B}(P)) + d_H(\mathcal{B}(P), \mathcal{T}^{\text{ref}}),$$

that can also be rewritten as:

$$d_H(K', \mathcal{T}^{\text{ref}}) \leq d_H(K', \mathcal{B}(P)) + \max_{K \in \mathcal{B}(P)} d_H(K, \mathcal{T}^{\text{ref}}).$$

For the relation $d_H(K', \mathcal{T}^{\text{ref}}) \leq \delta$ to be verified, it is sufficient to have:

$$d_H(K', \mathcal{B}(P)) + \max_{K \in \mathcal{B}(P)} d_H(K, \mathcal{T}^{\text{ref}}) \leq \delta.$$

The triangles of the ball $\mathcal{B}(P)$ being easily known, it is possible to explicitly evaluate the quantity $d_H(K', \mathcal{B}(P))$.

Let $K \in \mathcal{B}(P)$ and let $K' \in \mathcal{R}(Q)$ be two mesh triangles, we denote π_K (resp. $\pi_{K'}$) the orthogonal projection on K (resp. K') and then, the intersection $C_{K'}(K) = \pi_{K'}(K) \cap K'$ of the projection of K onto K' with K' is either the emptyset or a convex polygon, its vertices being denoted by S_i (cf. Fig. 3). Similarly, $C_K(K') = \pi_K(K') \cap K$ is a convex polygon (possibly reduced to the emptyset), its vertices being denoted by S'_m . Thus, we have:

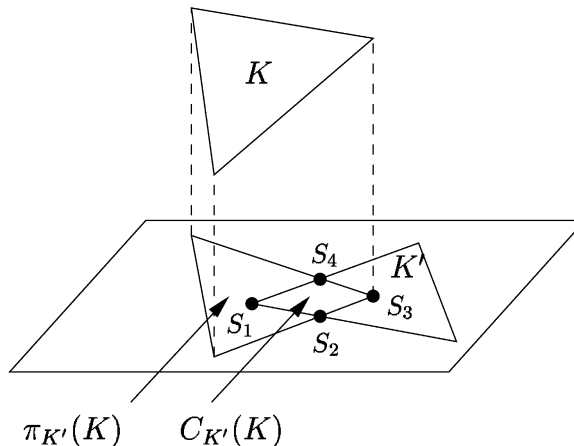


Fig. 3. The convex polygon corresponding to the region $C_{K'}(K) = \pi_{K'}(K) \cap K'$.

$$d(C_{K'}(K), \pi_{K'}^{-1}(C_{K'}(K))) = \begin{cases} \max_l d(S_l, \pi_{K'}^{-1}(S_l)) & \text{if } C_{K'}(K) \neq \emptyset, \\ 0 & \text{else,} \end{cases}$$

$$d(C_K(K'), \pi_K^{-1}(C_K(K'))) = \begin{cases} \max_m d(S'_m, \pi_K^{-1}(S'_m)) & \text{if } C_{K'}(K) \neq \emptyset, \\ 0 & \text{else.} \end{cases}$$

If we assume that the projection of $\mathcal{B}(P)$ on $\mathcal{R}(Q)$ is restricted (or equal) to the region $\mathcal{R}(Q)$:

$$\bigcup_{K' \in \mathcal{R}(Q)} \pi_{K'}(\mathcal{B}(P)) = \mathcal{R}(Q),$$

then, for each element $K' \in \mathcal{R}(Q)$ we obtain the two following covering up:

$$\begin{cases} K' = \bigcup_{K \in \mathcal{B}(P)} C_{K'}(K), \\ K' = \bigcup_{K \in \mathcal{B}(P)} \pi_K^{-1}(C_K(K')) \end{cases}$$

and thus, the Hausdorff distance from an element $K' \in \mathcal{R}(Q)$ to $\mathcal{B}(P)$ can be written as follows:

$$d_H(K', \mathcal{B}(P)) = \max_{K \in \mathcal{B}(P)} \delta(K, K'),$$

with

$$\delta(K, K') = \max(d(C_{K'}(K), \pi_{K'}^{-1}(C_{K'}(K))), d(C_K(K'), \pi_K^{-1}(C_K(K'))),$$

Notice that the Hausdorff distance between the ball $\mathcal{B}(P)$ and the region $\mathcal{R}(Q)$ can be expressed more simply as:

$$d_H(\mathcal{B}(P), \mathcal{R}(Q)) = \max_{K' \in \mathcal{R}(Q)} d(\pi_{K'}(P), P).$$

This new quantity is thus an upper bound of the distance $d_H(K', \mathcal{B}(P))$ and can be used to practically avoid the explicit computations of the regions $C_{K'}(K')$ and $C_{K'}(K)$. However, this bound does not allow to identify specifically the Hausdorff distance of each element K' to the ball $\mathcal{B}(P)$.

From a practical point of view, it is possible to find an approximation of the Hausdorff distance of each element $K' \in \mathcal{R}(Q)$ to $\mathcal{B}(P)$ by the Hausdorff distance from K' to the element \tilde{K} of $\mathcal{B}(P)$ sharing a common edge with K' (by definition, such an element always exists, cf. Fig. 4). It yields:

$$d_H(K', \mathcal{B}(P)) \approx d_H(K', \tilde{K}) = \max(d(P, \pi_{K'}(P)), d(Q, \pi_{K'}(Q)), d(P, Q)).$$

Hence, knowing the Hausdorff distances of the triangles created at iteration j to the reference mesh, it is possible to bound the distances of the new triangles, created at iteration $j + 1$. With each triangle K in \mathcal{T} is associated an upper bound $h(K)$ of its Hausdorff distance to the reference mesh \mathcal{T}^{ref} . For any newly created triangle K' of the mesh, we have:

$$h(K') = d_H(K', \mathcal{B}(P)) + \max_{K \in \mathcal{B}(P)} h(K).$$

A new triangle K' does not violate the geometry if, on the one hand:

$$h(K') \leq \delta$$

and if, on the other hand:

$$\forall k, \quad \langle v_k(K'), v(K') \rangle \geq \cos \theta,$$

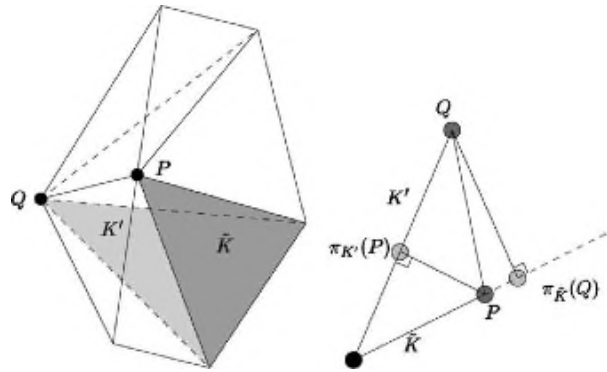


Fig. 4. Definition of the element \tilde{K} sharing a common edge with a given triangle K (left-hand side). Computation of the Hausdorff distance.

where $\nu(K')$ is the unit normal to triangle K' and $\nu_k(K')$ is the unit normal to the surface at the k th vertex of K' .

In addition, to control the mesh quality degradation, a coefficient β is introduced such that the mesh quality is considered as preserved if, for any new triangle:

$$q(K') \geq \beta \min_{K \in \mathcal{B}(P)} q(K),$$

where $q(\cdot)$ denotes the monotonous shape quality measure of a mesh element:

$$q(K) = c \frac{S(K)}{\sum_{e \in (K)} l^2(e(k))}.$$

$S(K)$ being the surface of K , $e(K)$ is an edge of K , $l(e(k))$ is the Euclidean length of edge $e(K)$ and $c = 4\sqrt{3}$, is a coefficient to have a quality equal to one (resp. zero) for an equilateral (resp. degenerated) triangle.

3.1.2. Edge flip

The other procedure of the mesh optimization process is a classical edge flip operation. It consists in replacing the two triangles sharing an edge with two new triangles, if the new configuration leads to an improvement of the element shape quality. This operation is carried out only if the two triangles considered are nearly coplanar (i.e., if the dihedral angle between the two triangles is close to 180°). In such a case, the geometry of the surface (the local curvature) is automatically preserved.

The two operations described so far (i.e., edge removal and edge flip) are usually followed by a node relocation procedure in order to, on the one hand, increase the number of edges that could potentially be removed and, on the other hand, improve the overall mesh quality.

3.1.3. Node relocation

Moving a mesh vertex consists in relocating this vertex, step by step, toward an “optimal” point location. Such a point corresponds to a configuration of optimally shaped (equilateral) triangles. To this end, at each mesh vertex, the surface can be locally approached by a quadric function passing at best through all neighboring vertices (following an idea suggested by [6]). We will now briefly indicate how to compute the optimal point location.

Let P be a mesh vertex, $\nu(P)$ be the unit normal to the surface at P and let $(\tau_1(P), \tau_2(P))$ be an orthonormal basis in the tangent plane to the surface at P , such that $(\tau_1(P), \tau_2(P), \nu(P))$ forms a direct basis in \mathbb{R}^3 (i.e., the local basis at P). Let consider the frame $(P, \tau_1(P), \tau_2(P), \nu(P))$ in \mathbb{R}^3 and let define in this frame the

quadric surface that locally approaches the surface at P . The quadric surface $\mathcal{Q}(P)$ is centered at $P = (0, 0, 0)$, thus, any point (x, y, z) of $\mathcal{Q}(P)$ is such that:

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

where a , b and c are the quadric coefficients that have to be numerically determined. The true surface is locally approached by a quadric surface, thus meaning that the neighboring vertices ($P_i = (x_i, y_i, z_i)$) of P shall belong at best to the quadric. This is equivalent to minimizing the squared distance of these points to the quadric surface. The coefficients a , b and c are then solutions of the following minimisation problem:

$$\min \sum_i (ax_i^2 + 2bx_iy_i + cy_i^2 - z_i)^2.$$

This problem has the same solution as the following linear system:

$$\begin{pmatrix} \sum_i x_i^4 & \sum_i 2x_i^3y_i & \sum_i x_i^2y_i^2 \\ \sum_i 2x_i^3y_i & \sum_i 4x_i^2y_i^2 & \sum_i 2x_iy_i^3 \\ \sum_i x_i^2y_i^2 & \sum_i 2x_iy_i^3 & \sum_i y_i^4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum_i x_i^2z_i \\ \sum_i 2x_iy_iz_i \\ \sum_i y_i^2z_i \end{pmatrix}.$$

The quadric surface $\mathcal{Q}(P)$ locally defined at vertex P will be used as geometric support to find the “optimal” position of P . Indeed, in a first stage, the point P^* leading to an optimal configuration (with respect to the element shapes) of the triangles of $\mathcal{B}(P)$ (the ball of P) is computed and then projected onto this quadric surface. The point P^* can be approached as the average location of the optimal points corresponding to equilateral triangles based on the boundary edges of $\mathcal{B}(P)$. The projection of P^* on the quadric surface $\mathcal{Q}(P)$ is the solution of the optimisation problem:

$$\min \|\vec{XP}\|^2 \quad \text{with } X \in \mathcal{Q}(P)$$

that can be reduced to solving a polynomial of order 5. This equation always has a real solution and, among the real solutions, we consider that (i.e., the point) closer to P^* . From a practical point of view, we use a Newton method for finding an approximation of the projection of P^* on the quadric surface. The corresponding algorithm can be written as [8]:

- set U_0 to P^* ;
- repeat $U_{k+1} = U_k - \frac{F(U_k)}{\|\nabla F(U_k)\|^2} \nabla F(U_k)$ (Newton step applied to the function $F(U_k + t\nabla F(U_k))$) until $\|U_{k+1} - U_k\|$ becomes sufficiently small.

At completion of the algorithm, the projection of P^* on the quadric surface is then point U_{k+1} .

It is also possible to consider the projection along the normal $v(P)$ of P^* on the quadric $\mathcal{Q}(P)$, that can be obtained more easily than the true projection of P^* on the quadric.

Actually, the point is moved if the surface geometry as well as the mesh quality are preserved. In such a case, the bounds on the Hausdorff distances of the newly created triangles (after the vertex coordinates have been updated) can be computed as in the previous section (i.e., similar to the edge flip case). In particular, the approximation of the Hausdorff distance can be applied according to Relation 3.1.1 where \mathcal{Q} represents the new position (after the point has moved) of P .

The proposed method enables to obtain a simplified mesh, associated with a given set of control parameters (δ, θ, β) , that allows to quantify, on the one hand, the desired level of geometric approximation and, on the other hand, the allowable mesh quality degradation. From a practical point of view, we consider a relaxation of the first two parameters and we carry out the mesh modifications in an iterative manner. The mesh simplification algorithm can be described as follows:

- set \mathcal{T} to \mathcal{T}^{ref} , Δ (resp. Θ) to $\delta_0 < \delta$ (resp. $\theta_0 < \theta$);
- while $\Delta \leq \delta$ and $\Theta \leq \theta$ do:
 - optimize all edges in \mathcal{T} ,
 - remove and optimize all edges in \mathcal{T} if the geometry (Δ, Θ) and the quality degradation β are preserved,
 - repeat the previous stage as long as \mathcal{T} is modified,
 - relocate the vertices of \mathcal{T} if the geometry (Δ, Θ) and the quality degradation β are preserved,
 - increment Δ and Θ .

Notice that it is also possible to sort the edges according to a criterion based on the mesh quality improvement. The suggested algorithm is linear in the number of the edges in the reference mesh, no specific order is considered for the edge analysis.

3.2. Extension to other surfaces

The extension of the method to the surfaces presenting geometric discontinuities consists in taking into account the interface curves (between two smooth pieces) as well as curves traced onto the surface. Notice that, in the first case, the variation of the normal to the surface is not continuous throughout the segments of interface curves. These critical curves are known in a discrete manner and are composed of edges of the reference mesh.

To enforce the proximity and regularity properties of the mesh with respect to the critical curves, two other tolerance regions are added to the already defined regions. The first region is a cylinder related to the global proximity, its principal axis corresponds to the critical curve and its radius is equal to a prescribed Hausdorff distance δ . The second property is enforced using another cone of regularity centered at each vertex of the critical curve in the reference mesh, its principal axis is given by the principal normal to the critical curve at this vertex and of (prescribed) aperture angle θ .

The deletion of an edge located along a critical curve Γ is “geometrically” validated if:

- the resulting configuration (i.e., the set of triangles) is at a Hausdorff distance δ from the reference mesh and the new critical edge (along Γ) is at a Hausdorff distance δ from Γ ;
- the normal to the new triangles are contained within the regularity cones at the vertices and the normal to the critical edge is contained within the cones associated with the edge endpoints.

Techniques close to that described in the previous section can also be used to compute the Hausdorff distances with respect to a critical curve.

Relocating a vertex along a critical curve consists in moving this vertex, step by step, toward an “optimal” point location, i.e., resulting in a configuration of optimal triangles. The new location is computed using a local approximation of the curve with a quadric curve passing at best through the adjacent vertices. As usual, the node is moved if the geometric approximation is preserved.

4. Validation

In this section, we propose a method to compute the distance between a reference mesh \mathcal{T}^{ref} of a surface and a simplified mesh \mathcal{T} of this surface. This procedure is a way of numerically validating the simplification approach described in the previous sections.

The main idea consists in computing the distance of any triangle in mesh \mathcal{T} to the reference mesh \mathcal{T}^{ref} . For the sake of simplicity, for a given triangle K of \mathcal{T} , we consider a sampling of points on K and we com-

pute the distance between K and \mathcal{T}^{ref} using the maximum of the distances between these points and \mathcal{T}^{ref} . Thus, the problem is reduced to the computation of the distance between a point in \mathbb{R}^3 and a mesh. To this end, for a given point of K , we use the following scheme:

1. find the closest vertex P_0 in \mathcal{T}^{ref} to the point,
2. find in $\mathcal{B}(P_0)$ the closest element K_0 to the point,
 - if K_0 is the closest element to the point in the ball of the vertices other than P_0 of K_0 , then the Hausdorff distance between the point and \mathcal{T}^{ref} is the distance between the point and K_0 ,
 - else, replace P_0 by the vertex P having a ball that contains the closest element to the point, go to step 2 and iterate.

In this algorithm, starting from the closest vertex to the sampling point, we progress in the balls of the vertices to find the closest element to the point. This is mainly related to the need to investigate the neighborhood of the current vertex in all directions (in particular, an analysis based on the edge neighborhood is not sufficient).

The distance between a point and a triangle in \mathcal{T}^{ref} can be computed using the barycentric coordinates λ_i , $i = 1, 3$ of the projection of this point onto the plane of the triangle:

- if $\lambda_i > 0$, $\forall i = 1, 3$, the distance is the distance between the point and its projection,
- if $\lambda_j < 0$, $\lambda_{i \neq j} > 0$, $\forall i = 1, 3$, the distance is the distance between the point and the element edge opposite to the vertex associated with λ_j ,
- if $\lambda_j > 0$, $\lambda_{i \neq j} < 0$, $\forall i = 1, 3$, the distance is the distance between the point and the vertex associated with λ_j .

The distance from a point to an edge is computed the same way the distance between a point and a triangle is computed: using the barycentric coordinates of the projection of the point and the line supporting the edge. To quickly and accurately find the closest vertex from the given sampling point, we take advantage of a bucket sort, its cells containing all vertices of mesh \mathcal{T}_{ref} .

This method provides the distance between \mathcal{T} and \mathcal{T}_{ref} , we proceed similarly to obtain the distance between \mathcal{T}_{ref} and \mathcal{T} .

5. Application examples

In this section, several application examples are provided to emphasize the efficiency of the proposed method. Each simplified mesh is associated to a pair of tolerance values, namely (δ, θ) . The δ tolerance is given as a percentage of the diagonal of the reference mesh bounding box. To measure the quality of the simplification, we give, for each simplification example, the following information:

- the rate of simplification with respect to the number of elements of reference mesh,
- the mean shape quality of the simplified mesh (a regular element has a shape quality equal to 1 and the quality of a poor element is close to 0),
- the gap map between the simplified mesh and the reference mesh,
- the total cpu time including the mesh simplification and the mesh validation on a HPJ5600/550 Mhz workstation.

5.1. A biomedical surface

The first example represents a polyhedral reconstruction of a biomedical iso-surface (a brain) from volumetric data. The resulting reference mesh is composed of 1 126 102 triangles (cf. Fig. 5) with a mean quality equal to 0.9. To reduce the apparent faceting due to the reconstruction process, a smoothing procedure without shrinkage [13] has been applied (cf. Fig. 6). Figs. 7 and 9 show two simplified meshes respectively related to tolerance pairs (0.35%, 33°) and (1%, 45°) which are generated (including the validation stage) in 301 and 231 s. The corresponding rates of simplification are 80% (227 624 elements) and 86.00% (165 886 elements). The mean quality of these simplified meshes are 0.83 and 0.79, respectively. Figs. 8 and 10 show the related gap map with respect of the reference meshes: the specified tolerance δ is satisfied for the simplified meshes.

5.2. A mechanical device

The second example concerns a wheel model provided by a commercial CAD modelling system. A reference mesh of the wheel is generated using a patch-dependent mesh generation algorithm (thus preserving the interface contours). It contains 161 840 elements (Fig. 11) with a mean quality of 0.63. Fig. 12 shows a simplified mesh related to the tolerance pair (0.2%, 33°) for which a maximum size mesh (3% of the diagonal of the reference mesh bounding box) has been prescribed. The mesh contains 20 064 elements corresponding to a simplification rate of about 88%. It has a mean mesh quality of 0.77 and has been generated and (validated) in 94 s. Fig. 13 shows the gap map between the simplified mesh and the reference one. Obviously, such a mesh can be used for a finite element computation. Fig. 14 shows another simplified mesh related to tolerance pair (0.5%, 36°) and contains 7 772 elements which corresponds to a simplification rate of 96%. The mesh has been generated and validated in 39 s, its mean quality is 0.66. Fig. 15 shows the

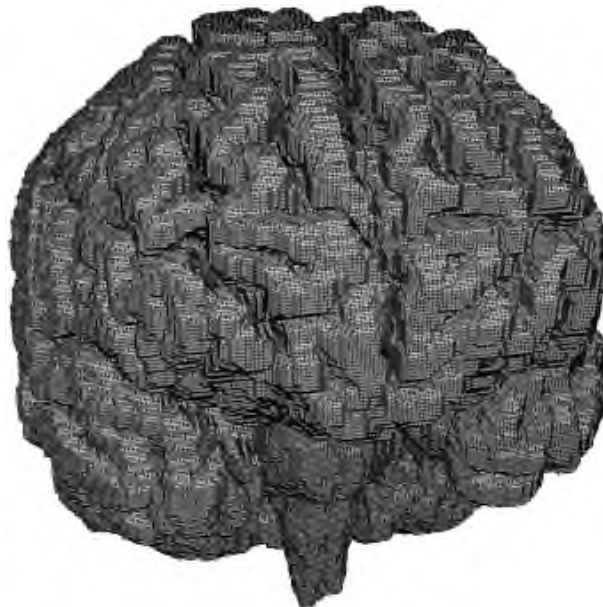


Fig. 5. Reference mesh of the brain.



Fig. 6. Reference mesh of the brain after the (non-shrinkage) smoothing procedure.

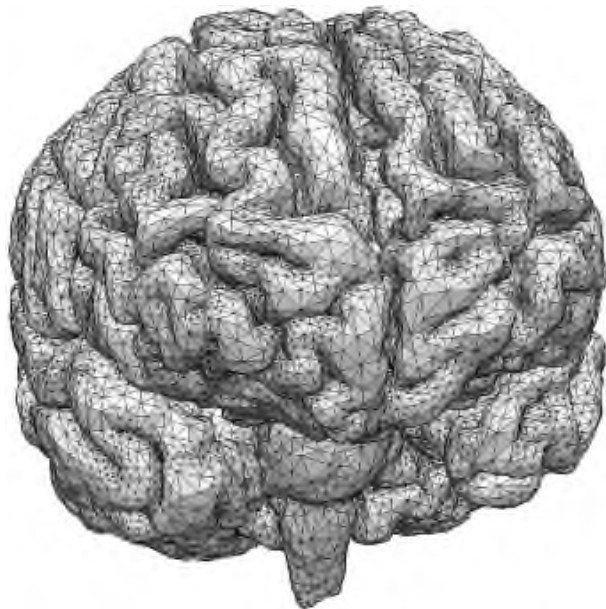


Fig. 7. Simplified mesh of the brain related to (0.35%, 33°).

gap map between this simplified mesh and the reference one. This new simplified mesh can be used for a fast prototyping of the wheel. One can easily notice that all the geometrical details are suppressed in the simplified meshes.

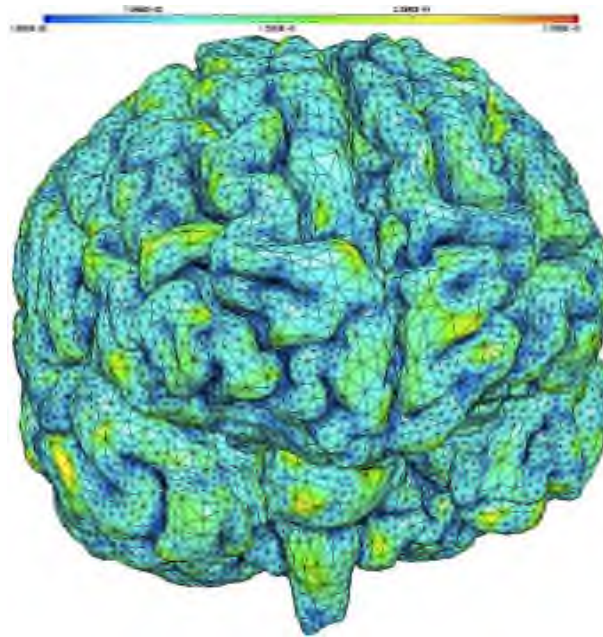


Fig. 8. Gap map related to (0.35%, 33°).



Fig. 9. Simplified mesh of the brain related to (1%, 45°).

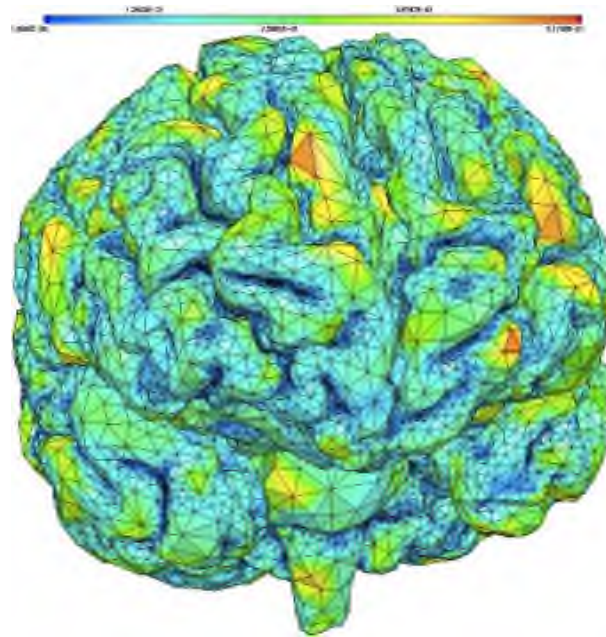


Fig. 10. Gap map related to (1%, 45°).

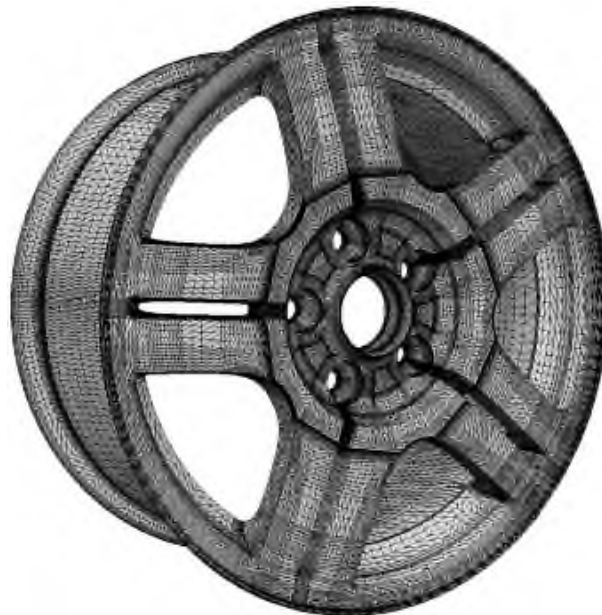


Fig. 11. Reference mesh of the wheel.

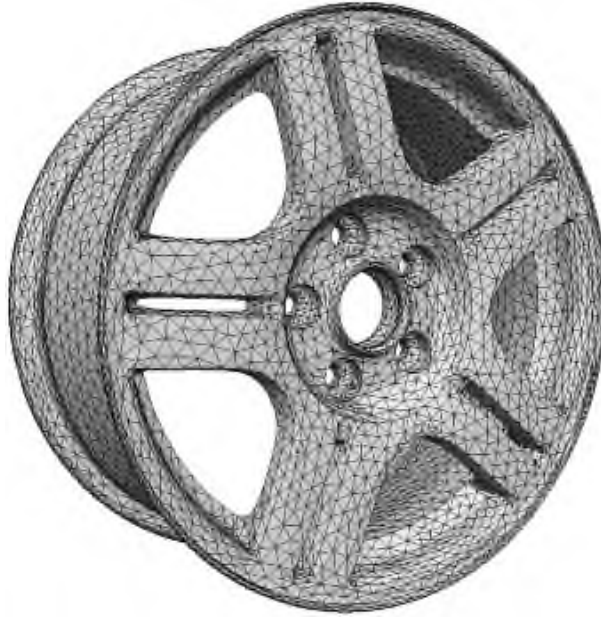


Fig. 12. Simplified mesh of the wheel related to (0.2%, 33°).

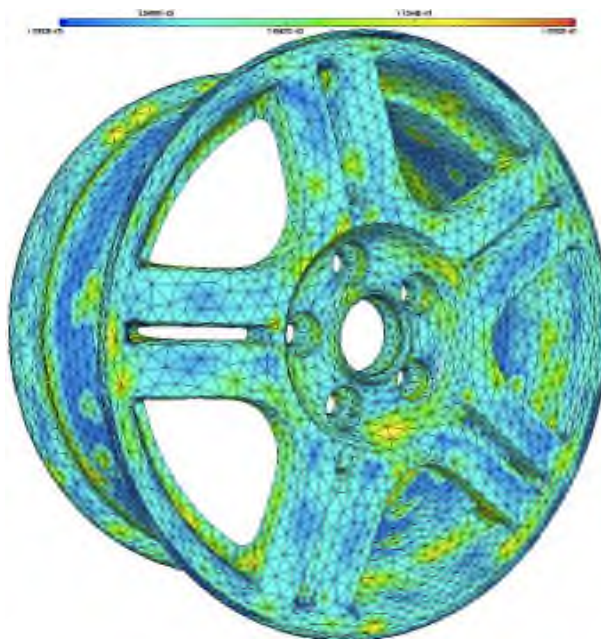


Fig. 13. Gap map related to (0.2%, 33°).

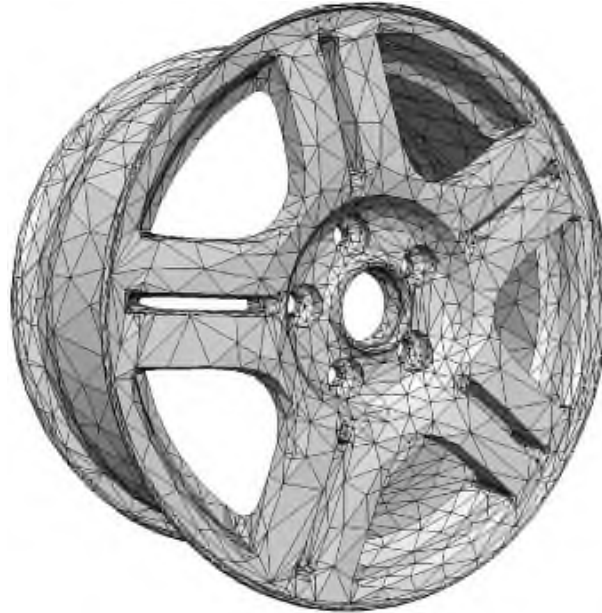


Fig. 14. Simplified mesh of the wheel related to (0.5%, 36°).

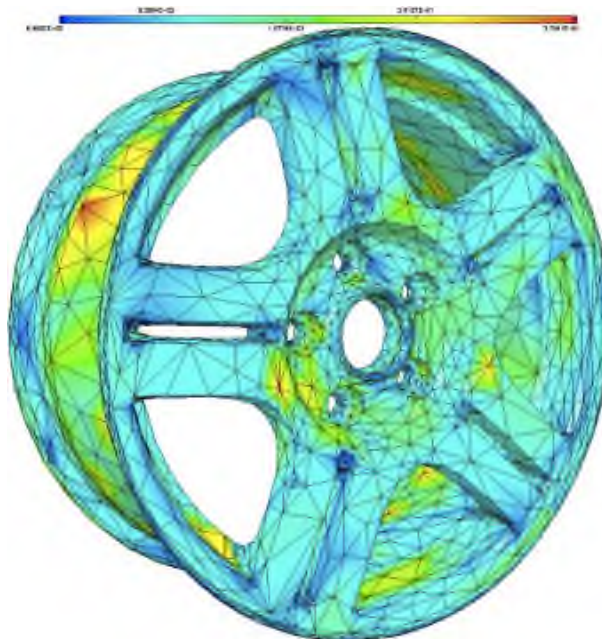


Fig. 15. Gap map related to (0.5%, 36°).

5.3. A very large statue model

The last example is a very fine mesh of an italian statue [11] containing 28055742 elements. This mesh is generated from a dense set of surface points obtained by a laser numerisation system. Fig. 16 shows a



Fig. 16. Simplified mesh of the statue related to (0.01%, 33°).



Fig. 17. Simplified mesh of the statue related to (0.02%, 33°).

shading of this statue from a simplified mesh containing 1041 126 elements (rate 0.96 of simplification) which is related to a tolerance pair of (0.01%, 33°).



Fig. 18. Simplified mesh of the statue related to (0.03%, 33°).



Fig. 19. Simplified mesh of the statue related to (0.05%, 33°).

Figs. 17–22 show simplified meshes of rate bigger that 0.99 (with an enlargement) related respectively to tolerance pairs of (0.02%, 33°), (0.03%, 33°), (0.05%, 33°), (0.1%, 33°), (0.2%, 33°) and (0.5%, 33°) with



Fig. 20. Simplified mesh of the statue related to (0.1%, 33°).



Fig. 21. Simplified mesh of the statue related to (0.2%, 33°).



Fig. 22. Simplified mesh of the statue related to (0.5%, 33°).

390 196, 238 734, 144 940, 114 626, 74 230 and 62 784 elements. These meshes are generated each in less than 1 h and are a mean quality of 0.9.

6. Conclusions

In this paper, we have described a new simplification method for surface meshes. Based on the Hausdorff distance, the proposed approach allows to explicitly control the distance to the original mesh. This is especially useful in most industrial problems in which this tolerance is a crucial parameter. We have validated this technique on several examples representative of various application fields. Large (high density) surface meshes have been successfully processed with this approach, giving satisfactory results in a reasonable cpu time.

The algorithm relies on a discrete approximation of the Hausdorff distance. A slightly finer analysis would lead to a more accurate formulation of this distance, also easier to implement. The proposed measure is based on a summation of the local Hausdorff distances, a noticeable improvement would consist in making this measure algebraic, based on the local geometric nature of the surface. The node relocation criterion has been designed so as to improve the overall element shape quality. A possible extension would be to take into account the surface geometry to increase the rate of simplification. Finally, a last challenge is envisaged: the simplification of hyper-dense surfaces, containing billions of points, that could potentially be resolved using this approach and parallel techniques.

References

- [1] F. Bernardini et al., Building a digital model of the Michelangelo's Florentine Pietà, *IEEE Comput. Graph. Appl.* 11 (2002) 59–67.

- [2] H. Borouchakiet, P.J. Frey, Maillage géométrique de surfaces. Partie II: appauvrissement, RR-INRIA 22 (1997) 3237.
- [3] J. Cohen, A. Varshnay, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, W. Wright, Simplification envelopes, in: Proc. Siggraph'96, 1996, pp. 119–128.
- [4] P.J. Frey, H. Borouchaki, Surface meshing using a geometric error estimate, *Int. J. Numer. Methods Engrg.* 58 (2) (2003) 227–245.
- [5] A. Guézic, Surface simplification inside a tolerance volume, IBM Research Report, RC-20440, 1996.
- [6] B. Hamann, Curvature approximation for triangulated surfaces, *Comput. Suppl.* 8 (1993) 139–153.
- [7] B. Hamann, A data reduction scheme for triangulated surfaces, *Comput. Aided Geom. Des.* 11 (1994) 197–214.
- [8] E. Hartmann, A marching method for the triangulation of surfaces, *Visual Comput.* 14 (1998) 95–108.
- [9] H. Hoppe, Progressive meshes, in: Proc. Siggraph'96, 1996, pp. 99–108.
- [10] A.D. Kalvin, R.H. Taylor, Superfaces: polygonal mesh simplification with bounded error, *IEEE Comput. Graph. Appl.* 11 (1996) 64–77.
- [11] M. Levoy et al., The digital Michelangelo Project: 3D scanning of large statues, in: Proc. ACM Siggraph, 2000.
- [12] W.J. Schroeder et al., Decimation of triangle meshes, *Comput. Graph. (Proc. ACM Siggraph)* 26 (2) (1992) 65–70.
- [13] G. Taubin, A signal processing approach to fair surface design, *Proc. Siggraph'95* 11 (1995) 351–358.
- [14] G. Turk, Re-tiling polygonal surfaces, *Comput. Graph.* 26 (2) (1992) 55–64.
- [15] P. Véron, J.C. Léon, Static polyhedron simplification using error measurements, *Comp. Aided Geom. Des.* 29 (4) (1997) 287–298.