Simultaneous Aligning and Smoothing of Surface Triangulations

J.M. Escobar, R. Montenegro, E. Rodríguez, and G. Montero

Institute for Intelligent Systems and Numerical Applications in Engineering, University of Las Palmas de Gran Canaria, Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain
{jmescobar,rmontenegro,erodriguez,gmontero}@siani.es
http://www.dca.iusiani.ulpgc.es/proyecto0507

Summary. In this work we develop a procedure to deform a given surface triangulation to obtain its alignment with interior curves. At present, we consider that these curves are defined by the orthogonal projection from plane cubic splines to the initial surface triangulation. For example, the curves can represent interfaces between different materials or boundary conditions, internal boundaries or feature lines. Another possibility of this procedure is the adaption of a reference mesh to changing curves in the course of an evolutionary process (for example, aligning of mesh nodes and edges to moving shocks in compressible flows). Specifically, we propose a new method that moves the nodes of the mesh, maintaining its topology, in order to achieve two objectives simultaneously: the piecewise approximation of the curves by edges of the surface triangulation and the optimization of the resulting mesh. We will designate this procedure as *projecting/smoothing* method and it is based on the smoothing technique that we have introduced for surface triangulations in previous works. The mesh quality improvement is obtained by an iterative process where each *free node* is moved to a new position that minimizes a certain objective function. The minimization process is done on a surface projection plane attending to the surface piece-wise approximation and to an algebraic quality measure (*mean ratio*) of the set of triangles that are connected to the *free node*. So, the 3-D local projecting/smoothing problem is reduced to a 2-D optimization problem. Several applications of this method are presented.

Keywords: Mesh alignment, Moving meshes, Mesh adaptation, Surface mesh smoothing, Node movement, R-adaptivity.

1 Introduction

The numerical simulation of physical problems requires the internal boundaries and discontinuities to be properly represented. Usually, the largest errors are introduced in a neighborhood of such discontinuities. These errors are often greatly reduced if the mesh is aligned with the discontinuities. That is why it is desirable to have a procedure capable of deforming a given triangulation to get its alignment with a curve. Although there are numerous works dealing with surface mesh optimization, see for example [9, 10], only a few of them address the problem of the exact mesh alignment with interior curves. In fact, the only

paper that we have found in the bibliography tackling this question in similar terms, but for quadrilateral grids is [12]. The authors consider the problem of aligning a planar grid with multiple embedded curves defined by basic segments as straight lines or arcs of circle. A different approach to the problem can be found in [3], where the curve is approximated by a polygonal line included in the surface triangulation, but in this case the segments are not edges of the mesh. The paper [19] presents a variant of Ruppert's algorithm for producing a 2-D Delaunay triangulation of a domain containing arbitrary curved inputs. Nevertheless, this algorithm does not allow a dynamical adaption of the mesh without remeshing.

The procedure that we describe in this paper aligns a given surface triangulation with an arbitrary curve and it is based on the surface mesh smoothing technique proposed in [5]. An analytical representation of the curve is not usually available. Instead, it is approximately known by a sequence of interpolating data points. We have chosen a parametric cubic spline as interpolating curve due it is C^2 continuous and it has other interesting properties that will be used later. Obviously, the grade of approximation of the curve depends on the element sizes, therefore, a good strategy is to combine the projecting/smoothing technique with a local mesh refinement [11]. Our procedure is specially indicated for evolutionary problems where the boundaries change their shape or position with time; for example, the ones related to fluid-structure interactions involving large displacement (see, for example [21]), or crack modeling. The projecting/smoothing technique could be also applied to domain decomposition, definition of material interfaces, free boundary problems, etc.

The organization of the paper is as follows. In section 2 a rough description of the proposed method is presented. In section 3 we propose an objective function, and the corresponding modification, able to untangle and smooth plane triangulations simultaneously. The projecting/smoothing method is initially analyzed for plane triangulations in section 4 and, afterward, it is extended to triangulations defined on curved surfaces in section 5. Section 6 is devoted to applications. The paper concludes with a brief discussion of the work and its possible extensions.

2 Statement of the Projecting/Smoothing Method

Let C be a curve, and suppose that it is embedded in a surface triangulation T_Σ (see Figure 1). The basic idea of the projecting/smoothing method lies in relocating the nodes of T_Σ closest to C in positions just sited in the curve. This operation, which we will refer to as node projection onto the curve, goes on until getting an approximate representation (interpolation) of C by linked edges of T_Σ. A node of T_Σ is considered *projectable* if we can displace it from its initial position to any point of C in such a way the local mesh does not get *tangled*. This projection implies an enforced alteration of the original positions of the nodes and, in general, has a negative effect on the quality of the triangles close

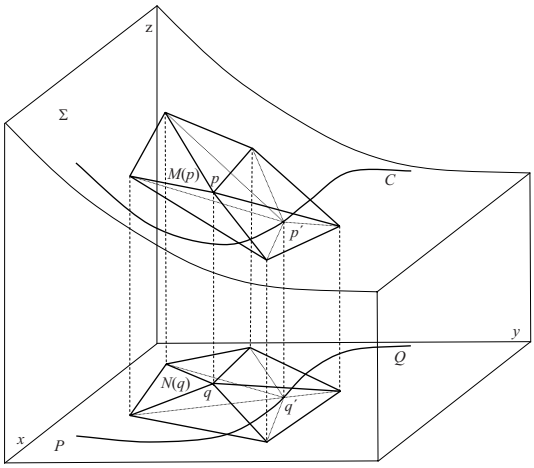


Fig. 1. The relocation of node $p \in \Sigma$ is performed in the plane P by projecting q on Q and, consequently, p on C

to C. To avoid this drawback, the remaining nodes are also displaced to new positions following the smoothing procedure proposed in [5].

For 2-D (or 3-D) meshes, the quality improvement can be obtained by an iterative process where each node of the mesh is moved to a new position that minimizes an objective function derived from certain *algebraic quality measure* of the local mesh [15, 8]. The objective function presents a barrier in the boundary of the *feasible region* associated to the free node. In this context the 2-D (or 3-D) feasible region is the set of points where the free node could be placed to get a *valid* local mesh, that is, without *inverted elements*. The barrier has an important role because it avoids the optimization algorithm to create a tangled mesh when it starts with a valid one. We show in [5] a procedure for smoothing surface triangulations taking into account these aspects. The basic idea lies in transforming the original problem on Σ into a two-dimensional one on a plane P. To do this, the local mesh $M(p)$, belonging to T_Σ, is orthogonally projected onto a plane P performing a local mesh $N(q)$, where p is the free node on Σ and q is its orthogonal projection onto P. The plane P is suitably chosen in terms of $M(p)$ in order to get a valid mesh on P (see Figure 1). Thus, the optimization of $M(p)$ is got by the appropriate optimization of $N(q)$. It involves the construction of *ideal* triangles in $N(q)$ that become near equilateral in $M(p)$.

When Σ is a curved surface, each triangle of $M(p)$ is placed on a different plane. Therefore, it is not possible to define the feasible region associated to the free node p. Nevertheless, the feasible region associated to node q is perfectly defined in plane P and it is denoted as $\mathcal{H}_q$. Furthermore, its associated objective function has a barrier at the boundary of $\mathcal{H}_q$ (see [6]). This is a crucial reason for working on P instead of on Σ.

In the present work the curve C is defined as the image of a curve Q sited on a plane P. Specifically, if we define a plane curve by the parametrization $Q(u) = (x(u), y(u))$ and we consider that $f(x, y)$ is the z coordinate of the underlaying surface (the true surface, if it is known, or the piece-wise linear interpolation, if it is not), then the curve C is given by $C(u) = (x(u), y(u), f(x(u), y(u)))$ (see Figure 1). We note that this type of parametrization can be straightforward introduced in the new *meccano* method which has been recently developed by the authors [4]. We remark also that, although the surface mesh smoothing process can be carried out in different planes chosen in terms of $M(p)$ [5], the particular way in which C is defined in the present paper demands a unique plane. A general parametric curve $C(u) = (x(u), y(u), z(u))$ will be considered in future works.

Since the problem of getting a piecewise approximation of C by edges of T_Σ is translated to the plane P, the task to determine if a node q can be projected onto Q and, that being the case, which is its optimal position, is undertaken by an objective function derived from algebraic quality measures of the local mesh $N(q)$. This objective function incorporates the modifications proposed in [6] in order to deal with tangled meshes. Obviously, a control of the allowed distance between $M(p)$ and $M(p')$ is done in the analysis of the movement of node q.

3 Smoothing and Untangling of Plane Triangulations

Firstly, we will focus our attention on finding an objective function to smooth a valid plane triangulation. As it is shown in [8], [13], and [14] we can derive optimization functions from algebraic quality measures of the elements belonging to a local mesh. Let us consider a triangular mesh T_P defined in $\mathbb{R}^2$ and let t be a triangle in the physical space whose vertices are given by $\mathbf{x}_k = (x_k, y_k)^T \in \mathbb{R}^2$, $k = 0, 1, 2$. To start with, we introduce an algebraic quality measure for t. Let t_R be the reference triangle with vertices $\mathbf{u}_0 = (0, 0)^T$, $\mathbf{u}_1 = (1, 0)^T$, and $\mathbf{u}_2 = (0, 1)^T$. If we choose $\mathbf{x}_0$ as the translation vector, the affine map that takes t_R to t is $\mathbf{x} = A\mathbf{u} + \mathbf{x}_0$, where A is the Jacobian matrix of the affine map referenced to node $\mathbf{x}_0$, given by $A = (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0)$. We will denote this type of affine maps as $t_R \xrightarrow{A} t$. Let now t_I be an *ideal* triangle whose vertices are $\mathbf{w}_k \in \mathbb{R}^2$, $(k = 0, 1, 2)$ and let $W_I = (\mathbf{w}_1 - \mathbf{w}_0, \mathbf{w}_2 - \mathbf{w}_0)$ be the Jacobian matrix, referenced to node $\mathbf{w}_0$, of the affine map $t_R \xrightarrow{W_I} t_I$; then, we define $S = AW_I^{-1}$ as the weighted Jacobian matrix of the affine map $t_I \xrightarrow{S} t$. In the particular case that t_I was the equilateral triangle t_E, the Jacobian matrix $W_I = W_E$ will be defined by $\mathbf{w}_0 = (0, 0)^T$, $\mathbf{w}_1 = (1, 0)^T$ and $\mathbf{w}_2 = (1/2, \sqrt{3}/2)^T$.

We can use matrix norms, determinant or trace of S to construct algebraic quality measures of t. For example, the Frobenius norm of S, defined by $|S| = \sqrt{tr\,(S^T S)}$, is specially indicated because it is easily computable. Thus, it is shown in [15] that $q_\eta = \frac{2\sigma}{|S|^2}$ is an algebraic quality measure of t, where $\sigma = \det(S)$. We use this quality measure to construct an objective function. Let $\mathbf{x} = (x, y)^T$ be the position vector of the free node q, and let S_m be the weighted

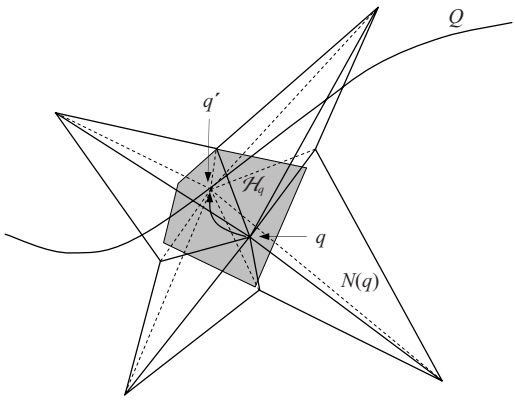


Fig. 2. The curve Q intersects the feasible region $\mathcal{H}_q$ (in gray) and, therefore, the node q is projectable, being q' its optimal position on the curve

Jacobian matrix of the m-th triangle of a valid local mesh $N(q)$ composed of M triangles, see Figure 2. The objective function associated to m-th triangle is $\frac{|S_m|^2}{2\sigma_m}$, and the corresponding objective function for the local mesh is

$$|K_\eta|_n(\mathbf{x}) = \left[\sum_{m=1}^{M}\left(\frac{|S_m|^2}{2\sigma_m}\right)^n(\mathbf{x})\right]^{\frac{1}{n}} \tag{1}$$

being n an integer number, typically $n = 1$ or $n = 2$.

The feasible region for the local mesh is defined as the interior of the polygonal set $\mathcal{H}_q = \bigcap_{m=1}^{M} H_m$, where H_m are the half-planes defined by $\sigma_m(\mathbf{x}) \geq 0$. We say that a triangle is *inverted* if $\sigma < 0$. The objective function (1) presents a barrier in the boundary of the feasible region. This barrier avoids the optimization method to create a tangled mesh when it starts with a valid one, but, on the other hand, it prevents the algorithm to untangle it when there are inverted elements. Therefore, this objective function is only appropriate to improve the quality of a valid mesh, not to untangle it. To construct an objective function applicable to deal with tangled meshes we propose to modify it following the criteria developed in [6]. This modification lies in substituting σ in (1) by the positive and increasing function

$$h(\sigma) = \frac{1}{2}(\sigma + \sqrt{\sigma^2 + 4\delta^2}) \tag{2}$$

where the parameter $\delta = h(0)$ is an appropriate small value.

In this way, the barrier associated with the singularities of $|K_\eta|_n(\mathbf{x})$ will be eliminated and the modified objective function will be smooth all over $\mathbb{R}^2$

$$\left|K'_\eta\right|_n(\mathbf{x}) = \left[\sum_{m=1}^{M}\left(\frac{|S_m|^2}{2h(\sigma_m)}\right)^n(\mathbf{x})\right]^{\frac{1}{n}} \tag{3}$$

This new objective function strongly penalizes the negative values of σ, so that the minimization process of (3) leads to the construction of a local mesh $N(q')$ without inverted triangles, provided it is possible. Note that the minimum of original and modified functions are nearly identical when $\mathcal{H}_q \neq \emptyset$ and δ tends to zero. With this approach, we can use any standard and efficient unconstrained optimization method to find the minimum of the modified objective function, see for example [2].

4 Alignment of Plane Triangulations

Node movement provides the mesh with the ability to align with an arbitrary curve. Suppose that Q is a curve defined on a 2-D triangulation T_P, our objective is to move some nodes of T_P, projecting them onto Q, to get an interpolation of Q by linked edges of T_P. To achieve this objective we have to decide which nodes of T_P can be projected onto Q without inverting any triangle of its local mesh. More accurately, we say that the free node q is projectable onto Q if there are points of this curve belonging to the feasible region $\mathcal{H}_q$ (see Figure 2).

In general, if q is projectable, its possible placement on Q is not unique. The projecting/smoothing method must determine if q can be projected onto Q and, if so, which is its optimal position. The last question can be answered by using the objective function (3) subject to the constrained $\mathbf{x} \in Q$. Thus, the problem of finding the optimal position to project the free node onto the curve is

$$\text{minimize}\left|K'_\eta\right|_n(\mathbf{x}),\ \text{subject to}\ \mathbf{x} \in Q \tag{4}$$

If $\bar{\mathbf{x}}$ is the position vector of the minimizing point q' of (4) and $\sigma(\bar{\mathbf{x}}) > 0$ for all triangle of $N(q')$, we conclude that q is projectable onto Q and $\bar{\mathbf{x}}$ is its optimal position. Otherwise, we say that node q is not projectable.

4.1 Curve Definition

The previous criterion allows us to determine whether q is projectable onto Q or not, but it involves a high computational cost because it needs to solve the constrained minimization problem (4). Nevertheless, it is clear that most nodes of T_P are not projectable because they are very far from any point of the curve. Therefore, it is convenient to have an efficient method to select those nodes, close to some segment of Q, expected to be projectable.

In many situations of practical interest we do not have an analytical representation of Q, but Q is approximately known by a sequence of interpolating data points. Among the options to define an interpolating curve, we have chosen a parametric cubic spline as it has many desired properties: it is a C^2 continuous

function, it has a very simple local form, small oscillations, etc. Moreover, each segment of the spline is a degree 3 Bézier curve that lies within the convex hull of its four defining control points (see, for example [1]). We will use this property in order to know if a given node is close to some segment of Q.

Let $\{P_0, P_1, \ldots, P_m\} \subset \mathbb{R}^2$ be a set of interpolating points belonging to plane P. The parametric cubic spline

$$Q(u) = (x(u), y(u)), \text{ where } u \in [u_0, u_m] \tag{5}$$

is an interpolating curve that satisfies $Q(u_i) = P_i$ for $i = 0, \ldots, m$ and two additional constraints in order to be fully defined. Usually, these constraints are imposed at the ends of the curve. For example, it is well known that the conditions $Q''(u_0) = 0$ and $Q''(u_m) = 0$ define a spline known as *natural*.

Every segment of the spline delimited by two consecutive interpolating points is a degree 3 polynomial. Suppose that $Q_i(t) = \mathbf{a}^i + \mathbf{b}^i t + \mathbf{c}^i t^2 + \mathbf{d}^i t^3$, with $\mathbf{a}^i$, $\mathbf{b}^i$, $\mathbf{c}^i$ and $\mathbf{d}^i$ in $\mathbb{R}^2$, is the polynomial associated to the segment Q_i $(i = 0, 1, \ldots, m-1)$ that runs from P_i to P_{i+1}, being $t \in [0, 1]$ the local parameter, see Figure 3. This one is related with the parameter of the entire curve by $t = (u - u_i) / (u_{i+1} - u_i)$.

4.2 Node Projection onto the Curve

The Q_i segment also is a degree 3 Bézier curve, given by $Q_i(t) = \sum_{j=0}^{3} \mathbf{u}_j^i B_j^i(t)$ with $t \in [0, 1]$, where $B_j^i(t)$ are the Berstein polynomials and $\mathbf{u}_j^i \in \mathbb{R}^2$ are the control points. The relation between the polynomial coefficients and the control points are given by

$$\begin{pmatrix} \mathbf{u}_0^i \\ \mathbf{u}_1^i \\ \mathbf{u}_2^i \\ \mathbf{u}_3^i \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 3 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 3 & 3 & 3 & 3 \end{pmatrix} \begin{pmatrix} \mathbf{a}^i \\ \mathbf{b}^i \\ \mathbf{c}^i \\ \mathbf{d}^i \end{pmatrix} \tag{6}$$

As we have already said, an interesting property of the Bézier curves establishes that the Q_i segment lies within the convex hull of its control points. If CH denotes the convex hull of a set of points, we have $Q_i \subseteq \mathrm{CH}\left(\mathbf{u}_0^i, \ldots, \mathbf{u}_3^i\right)$. Note that a necessary (but not sufficient) condition for the node q to be projectable onto Q is that its feasible region $\mathcal{H}_q$ intersects the convex hull of some segment of the curve. In other words, it must exist a segment Q_i such that $\mathcal{H}_q \cap \mathrm{CH}\left(\mathbf{u}_0^i, \ldots, \mathbf{u}_3^i\right) \neq \emptyset$. This property allows us to know beforehand which nodes are not projectable, because they yield an empty intersection for all segments of the curve. Nevertheless, calculating the set $\mathcal{H}_q$ and, moreover, its intersection with a convex set, is not a trivial problem, so it is more advisable to deal with a simplified version.

Let R_q and R_{Q_i} be the minimal rectangles, with sides parallel to the axes, enclosing the sets $N(q)$ and $\mathrm{CH}\left(\mathbf{u}_0^i, \ldots, \mathbf{u}_3^i\right)$, respectively. Then, due to $\mathcal{H}_q \subset R_q$, it is clear that q is projectable onto Q_i only if $R_q \cap R_{Q_i} \neq \emptyset$ (see Figure 3). The

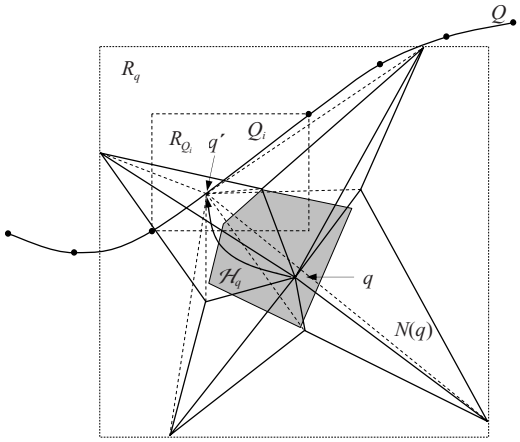


Fig. 3. The figure shows the situation in which $R_q \cap R_{Q_i} \neq \emptyset$, but node q is not projectable because the optimal position for the free node, q', is outside the feasible region

computation of this intersection allows us to take a quick decision about if a node is candidate to be projected onto the curve.

The algorithm to determine if q is projectable onto Q and, if it is so, which is its optimal position, can be summarized as follows. For each segment of the curve analyze $R_q \cap R_{Q_i}$ and, if this set is not empty, solve the minimization problem

$$\text{minimize} \left|K'_\eta\right|_n (Q_i(t)), \text{ for } t \in [0,1] \tag{7}$$

Let $\bar{t}$ be the global minimum of (7) and $\bar{\mathbf{x}}_i = Q_i(\bar{t})$ the corresponding position of the free node q on the segment Q_i. We say that $\bar{\mathbf{x}}_i$ is an *admissible* optimal position for the free node if $\sigma_m(\bar{\mathbf{x}}_i) > 0$ for $m = 1, \ldots, M$. In order to determine the optimal position of the free node on Q, we take $\bar{\mathbf{x}}_{opt}$ as the best admissible position for all segments Q_i. Obviously, if no admissible position exists, the conclusion of previous algorithm is that node q is not projectable onto Q.

Take into account that an admissible projection of a free node on Q can give rise to a local mesh with very poor quality. Although this effect is partly palliated after smoothing the remainder nodes, following the procedure described in section 3, it is appropriate to tighten the condition $\sigma_m(\bar{\mathbf{x}}_i) > 0$ enforcing $\sigma_m(\bar{\mathbf{x}}_i) > \varepsilon$, with $\varepsilon > 0$ a prescribed tolerance. Nevertheless, this more restrictive condition makes it difficult for the nodes to be projected onto the curve and it could produce situations where some sections of the curve are not interpolated by edges of T_P. This drawback will be studied in the next subsection but, for that purpose, it needs further clarification.

Up to now, we have accepted that parameter t pertains to the closed interval $[0,1]$ and, in consequence, the problem (7) admits a global minimum. But, with this consideration, the ends of the consecutive segments are shared and,

therefore, a projected point can belong to two segments at the same time. In order to avoid this ambiguity, we will assume that each segment $Q_i(t)$ is defined for $t \in [0, 1)$, except the last one, that it is for $t \in [0, 1]$ if the curve is open. In this way, each point of the curve belongs to a unique segment.

4.3 Discontinuities of the Mesh Alignment with the Curve

It can happen that, after repositioning all the nodes of the mesh, the piecewise approximation of Q by edges of T_P is not continuous. We can detect this discontinuity if we take into account that the projected nodes are arranged in the curve. Thus, a section of the interpolated curve among two consecutive projected nodes is discontinuous if they are not connected by an edge of T_P.

As the parameter $t \in [0, 1)$ induces an order relation in each segment of the curve and, in turn, each segment is ordered by its subindex, we can say that the node $p \in Q_i$ precedes $p' \in Q_j$ if $i < j$ or, in case of $i = j$, if the corresponding parameters satisfies $t_p < t_{p'}$. A possibility to correct a detected discontinuity in the piecewise approximation of Q is to relax the condition $\sigma_m(\bar{\mathbf{x}}_i) > \varepsilon$, by decreasing the value of ε. However, there are situations in which, even taking ε equal to zero, there are discontinuities impossible to avoid without removing some of the projected nodes. The Figure 4(a) shows a scheme of this problem. It can be seen that it is impossible to project the node q (neither r) without

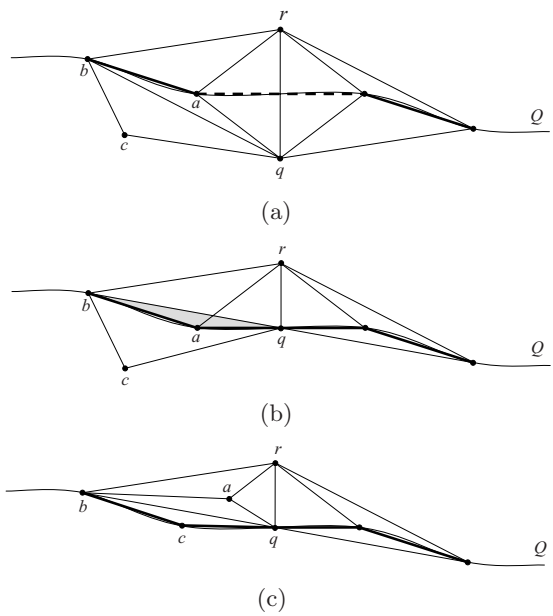


Fig. 4. The dashed line is non-recoverable without tangling the mesh (a). The free node q is enforced to be projected (b). The tangled triangle abq is untangled and the node c is also projected (c).

tangling the mesh. We propose a solution to this conflict by enforcing the free node q to be projected, even if a tangled mesh is created. The Figure 4(b) shows how the movement of q produces the tangled triangle abq.

Afterward, the position of q is fixed for subsequent iterations of the projecting/smoothing algorithm, but the surrounding nodes are free to move in search of their optimal positions that untangle the mesh and complete the interpolation of the curve (see 4(c)). So, the algorithm extracts nodes from the curve if their current positions are not admissible (see the new position of node a in figure 4(c)).

Sometimes the curve represented by splines has sharp features that we want to preserve in the piecewise interpolation. To reach this objective we select, from the interpolating points, a set of *prescribed* points sited in strategic locations. Once the projecting/smoothing process has finished, the algorithm searches among the nodes projected on Q, which one is the optimal candidate, say q, to be relocated in the position of each prescribed node. If $\mathbf{x}_{pres}$ is the position of certain prescribed point, the node q is chosen, among the nodes projected on Q and close to $\mathbf{x}_{pres}$, as the one that maximizes the quality of $N(q)$ when q is enforced to take the position $\mathbf{x}_{pres}$. Obviously, if $N(q)$ is not valid after the relocation of q, a new iteration of the projecting/smoothing procedure must be done.

5 Extension to Curved Surfaces

We are interested in extending the projecting/smoothing method to curved surfaces. As we pointed in section 2, the original problem on Σ is transformed into another one on the plane P. The more significant difference with respect to the former method lies in searching *ideal* triangles in $N(q)$ that become equilateral in $M(p)$. Due to restriction of the extension of this paper, we summarize in this section our main result relative to this aspect. A detailed analysis can be carried out in the surface mesh smoothing procedure that is presented in [5]. Its connection with the problem of surface mesh aligning with curves could be easily implemented.

Consider that triangle $t \in N(q)$ (located in plane P) is the orthogonal projection of triangle $\tau \in M(p)$ (located in plane π), see Figure 5. Let t_R and τ_R be the references triangles defined in planes P and π, respectively. Suppose that we chose as ideal triangle in π the equilateral one ($\tau_I = \tau_E$). Our goal is to find the *ideal* triangle $t_I \subset P$, moving q on P, such that t_I is mapped into an equilateral one, $\tau_E \subset \pi$. For this purpose, the following similarity transformation between the matrices S and S_I was proved in [5]

$$S = S_E^{-1} S_I S_E \tag{8}$$

where S is defined on the plane π as the 2×2 weighted Jacobian matrix of the affine map that transforms the equilateral triangle into the physical one, that is, $\tau_E \xrightarrow{S} \tau$, S_E is defined on plane P as the *equilateral-weighted* Jacobian matrix

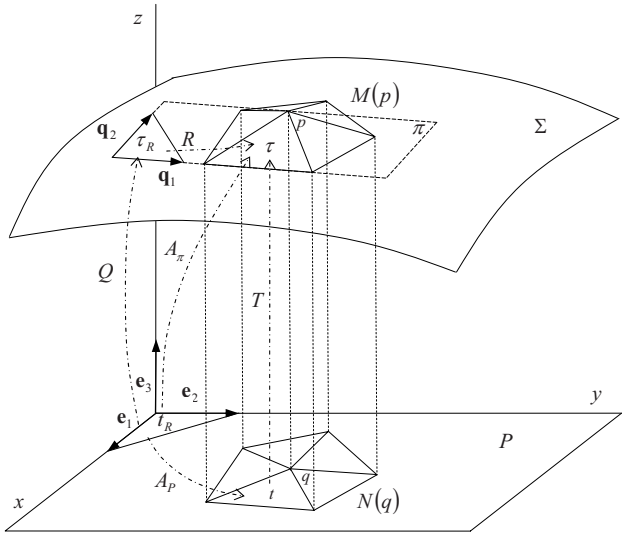


Fig. 5. Local surface mesh, $M(p)$, and its projection, $N(q)$, on plane P

of the affine map $t_E \overset{S_E}{\to} t$ and S_I is defined on plane P as the *ideal-weighted* Jacobian matrix of the affine map $t_I \overset{S_I}{\to} t$.

Matrices S may be used, as it is defined in (8) for any triangle $\tau \in M(p)$, to construct the objective function associated to $M(p)$ and, then, solve the optimization problem. Nevertheless, this procedure has important disadvantages. First, the optimization of $M(p)$, working on the true surface, would require the imposition of the constraint $p \in \Sigma$. It would complicate the resolution of the problem because, in many cases, Σ is not defined by a smooth function. Moreover, when the local mesh $M(p)$ is on a curved surface, each triangle is sited on a different plane and the objective function, constructed from S, lacks barriers. It is impossible to define a feasible region in the same way as it was done at section 3. Indeed, all the positions of the free node, except those that make $\det(S) = 0$ for any triangle, produce correct triangulations of $M(p)$. However, there are another unacceptable positions of the free node.

To overcome these difficulties we proposed in [5] to carry out the optimization of $M(p)$ in an indirect way, working on $N(q)$. With this approach the movement of the free node will be restricted to $\mathcal{H}_q \subset P$, which avoids unacceptable surface triangulations to be formed. Then, the original smoothing problem is transformed in a two-dimensional approach on P. The algorithm to determine the optimal projection of a free node $p \in \Sigma$ onto the curve C is reduced to the one presented in section 4 for reaching the optimal projection of q onto Q. If $\bar{\mathbf{x}}_{opt}$ is the optimal projection of the free node q onto Q, then the corresponding position on the surface is given by $\bar{\mathbf{y}}_{opt} = (\bar{\mathbf{x}}_{opt}, f(\bar{\mathbf{x}}_{opt}))^T$, where $f(\mathbf{x})$ is the z coordinate of the underlaying surface. If this one is not known analytically,

we take the initial triangulation as reference surface. The algorithm follows the usual smoothing procedure when the free node is not projectable on C. The discontinuities of the mesh alignment with the curve are solved by using the same idea of section 4.3.

In order to prevent a loss of the details of the original geometry when we are smoothing the mesh, our algorithm evaluates the difference of heights ($[\Delta z]$) between the centroid of the triangles of $M(p)$ and the reference surface, every time a new position of the free node is calculated. If this distance exceeds a threshold, $\Delta(p)$, the movement of the node is aborted and the previous position is stored. This threshold $\Delta(p)$ is established attending to the size of the elements of $M(p)$. That is, the algorithm evaluates the average distance between the free node and the nodes connected to it, and takes $\Delta(p)$ as percentage of this distance. Other possibility is to fix $\Delta(p)$ as a constant for all local meshes. In the particular case in which we have an explicit representation of the surface by a function $f(x, y)$, $\Delta(p)$ can be established as a percentage of the maximum difference of heights between the original surface and the initial mesh.

6 Applications

In this section we present an application that shows the satisfactory behavior of the projecting/smoothing technique. In particular, we have applied the projecting/smoothing technique to the Igea surface triangulation (see Figure 6) obtained from *http://www.cyberware.com/*. The mesh contains 67170 triangles and 33587 nodes.

Our goal is to obtain a new triangulation (maintaining the initial mesh topology) after applying the projecting/smoothing procedure to reach the alignment of the new mesh with the contours of a mask and of a star drawn on the face of Igea. These curves are defined by the spline piece-wise interpolation of a few points that are placed on a front view of Igea. In Figure 7 we show the polylines that connect the fixed points for the definition of the mask and the star. Moreover, we construct several reference windows to define surface patches and to evaluate the coordinate transformation from the image parametric space to the physical one. In order to keep the sharp angles of the star drawing, we have prescribed 10 points as the extremes of 10 splines. We propose the following strategy to get our objective.

Initially, we apply the smoothing technique [5] to the whole triangulation. In this case, the projection plane is chosen in terms of the local mesh to be optimized. The resulting mesh, after 4 iterations of our optimization procedure, is shown in Figure 8. The value of the average mesh quality (measured with the algebraic quality metric based on the condition number proposed in [8]) increases from 0.794 to 0.913. A more significant data is that average quality of the worst 100 triangles increases from 0.379 to 0.575. We have fixed $\Delta(p)$ as 10% of average distance between the free node and the nodes connected to it. More details about this application can be seen in [17].

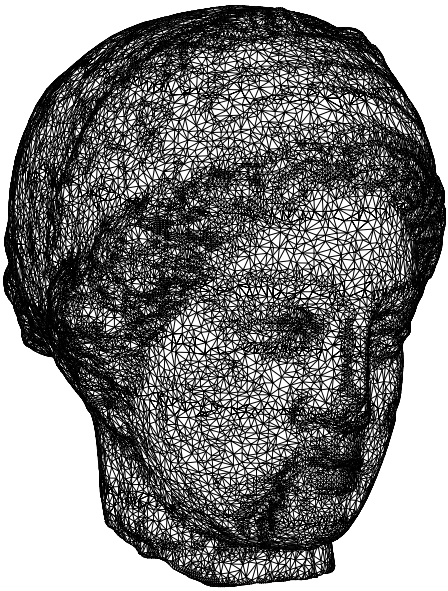


Fig. 6. Original mesh of Igea obtained from *http://www.cyberware.com/*

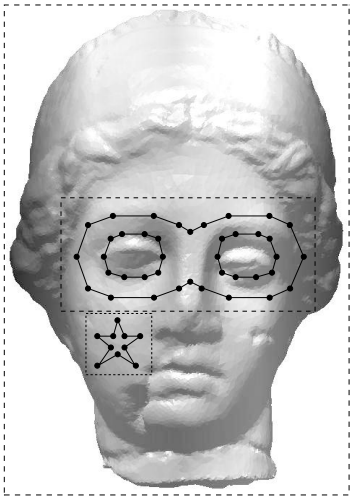


Fig. 7. Point input data for the definition of the curves, approximation of the splines as polylines and reference windows

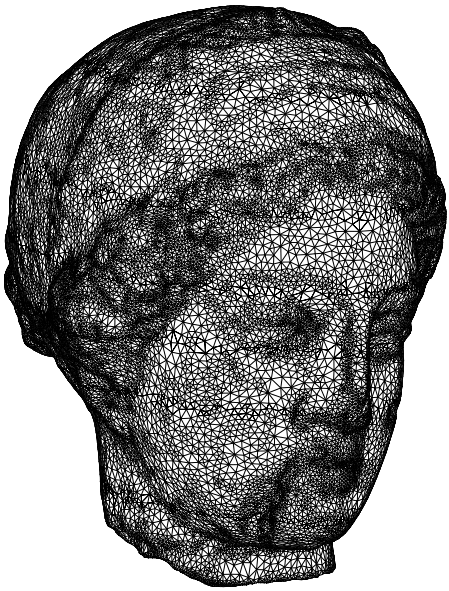


Fig. 8. Optimized mesh of Igea after 4 iterations of our smoothing procedure

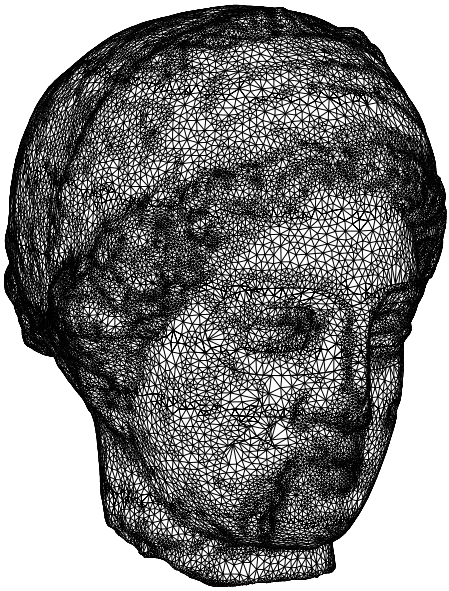
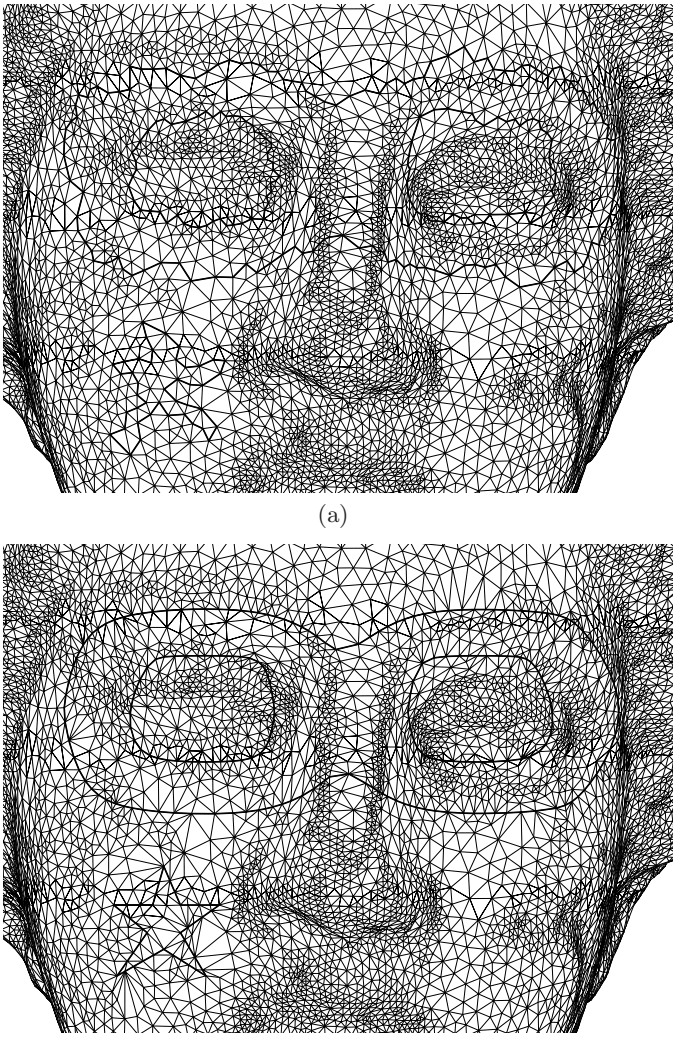


Fig. 9. Aligned and optimized mesh of Igea after 12 iterations of the local projecting/smoothing procedure



(a)

(b)

Fig. 10. Detail of the initial mesh of Igea with marked edges before projection (a). The same edges are remarked after the algorithm has projected them onto the contours of the mask and the star (b).

In order to reduce the computational cost of the alignment step, the projecting/smoothing process is carried out on the surface patches associated to the mask and to the star. For this purpose, we select the set of triangles whose centroids are included in the reference windows of the mask and of the star. Then, we apply the projecting/smoothing procedure to these two sets separately. We

note that, in the former process, the boundary of each patch triangulation is fixed. So, we obtain an appropriate connection of the modified local meshes with the rest of the surface triangulation. These ideas could be used for a parallel implementation of the simultaneous aligning and smoothing local technique.

In Figure 9 it is presented a general view of the surface triangulation of Igea after applying 12 iterations of the aligning and smoothing procedure. The approximation of the contours by edges of the resulting triangulation is marked. After the application of our algorithm the values of minimum and average qualities become 0.100 (same value than in previous meshes) and 0.911, respectively. The average quality of the worst 100 triangles is 0.519. Therefore, the mesh qualities are similar before and after the application of the projecting/smoothing technique.

Two details of the initial and final meshes (Figures 8 and 9) are shown in Figures 10(a) and (b), respectively. We represent the same marked edges before and after the local projecting/smoothing process. We note that it is very difficult to determine *a priori* which are the best edges for a suitable approximation of the contours of the mask and the star. However, the algorithm finds appropriate nodes (and consequently edges) automatically.

7 Concluding Remarks and Future Research

In this paper we have introduced the projecting/smoothing technique which is able to align a surface triangulation with arbitrary curves without producing, in general, a significant decrease in the minimum quality of the mesh. Indeed, the average quality is increased in many cases as the remainder part of the mesh undergoes a smoothing process.

In present work the curves have been defined by splines whose interpolating points are fixed on a plane. Applications of this technique can be done in a straightforward manner, for example, in environmental modeling [16, 18] for aligning topographic surface meshes to significant contours, as coastlines, river banks, etc. In addition, this particular curve definition can be applied on different patches of a more general surface.

Our method for aligning and smoothing of surface triangulations could be generalized by using a global parametric space (in similar terms as it is proposed in [7, 20]) which makes the projection on a plane unnecessary. So, general parametric curves embedded on the surface will be considered. Another more ambitious generalization lies in extending the present method to align a tetrahedral mesh with interior surfaces. This is an open problem. It is clear that the mesh alignment problem is not always possible to solve. Generally, the existence of an admissible solution can not be assured. Moreover, several admissible solutions may exit. Obviously, the existence of solution of the problem depends on size, quality and topology of the initial mesh and regularity of the embedded curves or surfaces.

Acknowledgments

This work has been supported by the *Secretaría de Estado de Universidades e Investigación* of the *Ministerio de Educación y Ciencia* of the Spanish Government and FEDER, grant contracts: CGL2007-65680-C03-01 and CGL2008-06003-C03-01.

References

1. Bartels, H.R., Beatty, J.C., Barsky, B.: An introduction to Splines for use in computer graphics & geometric modeling. Morgan Kaufmann, Los Altos (1987)
2. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear programing: theory and algorithms. John Wiley and Sons Inc., New York (1993)
3. Bonneau, G.P., Hahmann, S.: Smooth polylines on polygon meshes. In: Brunnett, G., Hamann, B., Mueller, H. (eds.) Geometric modeling for scientific visualization, pp. 69–84. Springer, Berlin (2003)
4. Cascón, J.M., Montenegro, R., Escobar, J.M., Rodríguez, E., Montero, G.: A new *meccano* technique for adaptive 3-D triangulations. In: Proc 16th Int Meshing Roundtable, pp. 103–120. Springer, Berlin (2007)
5. Escobar, J.M., Montero, G., Montenegro, R., Rodríguez, E.: An algebraic method for smoothing surface triangulations on a local parametric space. Int. J. Num. Meth. Eng. 66, 740–760 (2006)
6. Escobar, J.M., Rodríguez, E., Montenegro, R., Montero, G., González-Yuste, J.M.: Simultaneous untangling and smoothing of tetrahedral meshes. Comput. Meth. Appl. Mech. Eng. 192, 2775–2787 (2003)
7. Floater, M.S.: Parametrization and smooth approximation of surface triangulations. Comput. Aid. Geom. Design 14, 231–250 (1997)
8. Freitag, L.A., Knupp, P.M.: Tetrahedral mesh improvement via optimization of the element condition number. Int. J. Num. Meth. Eng. 53, 1377–1391 (2002)
9. Frey, P.J., Borouchaki, H.: Geometric surface mesh optimization. Comput. Visual Sci. 1, 113–121 (1998)
10. Garimella, R.V., Shaskov, M.J., Knupp, P.M.: Triangular and quadrilateral surface mesh quality optimization using local parametrization. Comput. Meth. Appl. Mech. Eng. 193, 913–928 (2004)
11. González-Yuste, J.M., Montenegro, R., Escobar, J.M., Montero, G., Rodríguez, E.: Local refinement of 3-D triangulations using object-oriented methods. Adv. Eng. Soft. 35, 693–702 (2004)
12. Hyman, J.M., Li, S., Knupp, P.M., Shashkov, M.: An algorithm to align a quadrilateral grid with internal boundaries. J. Comput. Phys. 163, 133–149 (2000)
13. Knupp, P.M.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I - A framework for surface mesh optimization. Int. J. Num. Meth. Eng. 48, 401–420 (2000)
14. Knupp, P.M.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II - A framework for volume mesh optimization and the condition number of the Jacobian matrix. Int. J. Num. Meth. Eng. 48, 1165–1185 (2000)
15. Knupp, P.M.: Algebraic mesh quality metrics. SIAM J. Sci. Comput. 23, 193–218 (2001)

16. Montenegro, R., Montero, G., Escobar, J.M., Rodríguez, E., González-Yuste, J.M.: Tetrahedral mesh generation for environmental problems over complex terrains. In: Sloot, P.M.A., Tan, C.J.K., Dongarra, J., Hoekstra, A.G. (eds.) ICCS-ComputSci 2002. LNCS, vol. 2329, pp. 335–344. Springer, Heidelberg (2002)
17. Montenegro, R., Escobar, J.M., Montero, G., Rodríguez, E.: Quality improvement of surface triangulations. In: Proc 14th Int Meshing Roundtable, pp. 469–484. Springer, Berlin (2005)
18. Montero, G., Rodríguez, E., Montenegro, R., Escobar, J.M., González-Yuste, J.M.: Genetic algorithms for an improved parameter estimation with local refinement of tetrahedral meshes in a wind model. Adv. Eng. Soft. 36, 3–10 (2005)
19. Pav, S.E., Walkington, N.J.: Delaunay refinement by corner looping. In: Proc 14th Int Meshing Roundtable, pp. 165–181. Springer, Berlin (2005)
20. Sheffer, A., De Sturler, E.: Smoothing an overlay grid to minimize linear distortion in texture mapping. ACM Trans. Graph 21, 874–890 (2002)
21. Stein, K., Tezduyar, T.E., Benney, R.: Automatic mesh update with the solid-extension mesh moving technique. Comput. Meth. Appl. Mech. Eng. 193, 2019–2032 (2004)