Filling holes in meshes using a mechanical model to simulate the curvature variation minimization

Jean-Philippe Pernot,* George Moraru and Philippe Véron
LSIS – UMR CNRS 6168
CER ENSAM, 2 cours des Arts et Métiers, 13617 Aix-en-Provence cedex, France
email: {*jean-philippe.pernot,george.moraru,philippe.veron*}*@aix.ensam.fr*

Abstract

The presence of holes in a triangle mesh is classically ascribed to the deficiencies of the point cloud acquired from a physical object to be reverse engineered. This lack of information results from both the scanning process and the object complexity. The consequences are simply not acceptable in many application domains (e.g. visualisation, finite element analysis or STL prototyping). This paper addresses the way these holes can be filled in while minimizing the curvature variation between the surrounding and inserted meshes. The curvature variation is simulated by the variation between external forces applied to the nodes of a linear mechanical model coupled to the meshes. The functional to be minimized is quadratic and a set of geometric constraints can be added to further shape the inserted mesh. In addition, a complete cleaning toolbox is proposed to remove degenerated and badly oriented triangles resulting from the scanning process.

Keywords: Reverse Engineering, geometric modelling, holes in meshes, triangle mesh deformation, curvature variation minimization, linear mechanical model, shape manipulations.

1 Introduction

Reverse Engineering is a powerful technique used to create a digital representation of an existing physical object. The reconstruction process starts with the acquisition of a point cloud from the outer surface of the physical object. A triangle mesh can be created from these dense unorganized data points [14, 12, 2, 4]. B-Spline/NURBS surfaces may also be created either directly from the point cloud or from its triangulation [20, 13, 11]. More recently, the use of subdivision surfaces for surface fitting and surface reconstruction has also been explored [21, 17, 30]. Here again, most of the proposed approaches begin with a triangulation of the point cloud. As a consequence, it is crucial to find a triangle mesh that best fits the object outer surface.
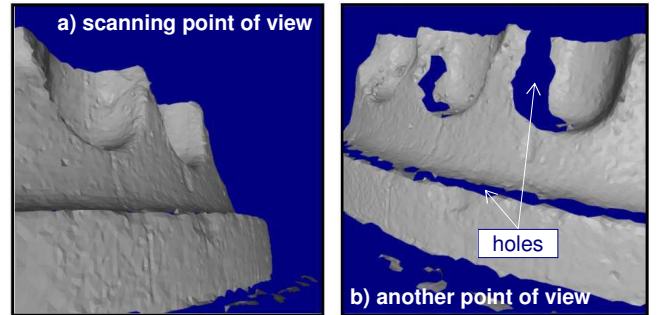

Figure 1: Example of holes (b) resulting from the acquisition of an object from a single point of view (a).

Depending on both the complexity of the object to be reverse engineered and the adopted data acquisition system technology (e.g. Coordinate Measuring Machines or laser scanning), some areas of the object outer surface may never be accessible. Figure 1 shows the result of an acquisition with a laser from a single point of view. Using the point of view (a), the scanner is unable to reach some portions of the object visible from another point of view (b). This induces some deficiencies in the point cloud and a set of holes in the triangle mesh. This is not acceptable. The presence of undesired holes may induce unexpected results when doing rapid prototyping or finite element analysis for example.

In this paper, we propose a set of models, methods and tools to fill in undesired holes in meshes. The filling process acts in several steps illustrated on the basic example of figure 2. First, the hole contour is identified (a) and cleaned (b) to remove badly oriented and degenerated triangles due to the scanner noise (section 4). A topological grid is then inserted (c) to fill in the hole (section 5). Finally, the inserted mesh is deformed (d) to satisfy various blending criteria with the initial mesh (section 6). We are notably able to minimize the curvature variation across the hole contour while solving a linear equations system. The curvature variation is simulated by the variation between external forces applied to the nodes of a linear mechanical model coupled to the meshes. The functional to be minimized is quadratic and

---

*corresponding author: jean-philippe.pernot@aix.ensam.fr

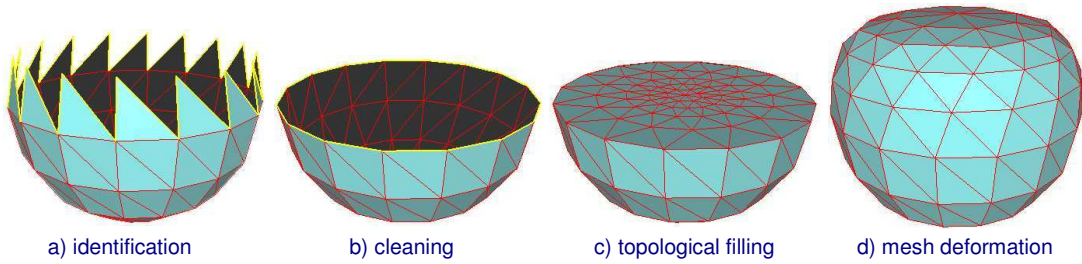| a) identification | b) cleaning | c) topological filling | d) mesh deformation |

Figure 2: Overall filling process on a simple example: after the identification and cleaning of the hole contour, the missing area is filled in with a topological grid whose final shape results from a deformation process.

a set of geometric constraints can be added to further shape the inserted mesh. These are strong improvements with respect to the existing techniques found in the literature (section 2). The filling process is illustrated with results produced by our prototype software (section 7). The limits and possible upgrades are finally discussed in section 8.

2   Related work

Various techniques have been proposed to fill in undesired holes in meshes. Two main categories can be distinguished: the geometric and non-geometric approaches.

Among the non-geometric approaches, [8] use a volumetric representation to detect the mesh areas that have to be filled in. [9] apply a volumetric diffusion process to extend a signed distance function through this volumetric representation until its zero set bridges whatever holes may be present. This iterative approach is particularly well adapted for complex geometrical and topological holes. Unfortunately, it does not ensure that the inserted mesh smoothly vanish on the surrounding mesh. A similar approach has been developed by [22] for the simplification and the repairing of polygonal meshes. [32] also represent the surface of interest in implicit form, and fill in the holes with a system of geometric partial differential equations derived from image inpainting algorithms. These equations are based on the geometric characteristics of the known mesh (e.g. the curvatures) and are applied only at the holes and a neighbourhood of them. Being these equations anisotropic and geometry based, they lead to a slightly slower algorithm than the one of [9]. [7] use a finite element method to minimize the integral of the squared mean curvature (the so-called Willmore energy) of the filled hole. Their process is iterative and can only ensure a tangency continuity with the surrounding mesh.

Considering the geometric approaches, [19] proposes a filling process quite similar to our (fig. 2). The hole is first detected and filled in with a minimum area triangulation of its 3D contour [3]. The triangulation is refined so that the triangle density agrees with the density of the surrounding mesh triangles [26]. The filled hole is finally smooth with a fairing technique based on an umbrella operator [16]. The faired mesh is obtained while solving a system of linear equations. It solely satisfies tangency blending conditions with the initial mesh. [28] propose a fairing technique based on solving a non-linear fourth order partial differential equation. G^1 boundary conditions are satisfied but the resolution is iterative. Also the implicit fairing of [10] requires an iterative resolution process. Other approaches are based on the Moving Least Squares projection which induces a non-linear minimization process [34, 31].

To conclude, the main drawbacks of most of these approaches concern the use of iterative resolution processes to find the shape of the inner mesh and/or the non satisfaction of curvature blending conditions between the inner and surrounding meshes. In fact, most of these approaches do not use enough geometric information available on the surrounding mesh. This is not true for the context-based surface completion algorithm of [29] where the inner mesh is defined according to shapes that can be very far from the hole. But their process may lead to very unexpected results. Finally, most of the processes do not enable the prescription of additional constraints inside the inner mesh. These are the limits we try to overcome. We will discuss how successful we are in section 8.

3   Terminology and hypothesizes

A triangle mesh is defined by a set of oriented triangles joining a set of vertices. Two triangles are adjacent if they share a common edge. Adjacent triangles are consistently oriented if their common edge is covered in opposite directions. A mesh is oriented if all its triangles are consistently oriented. An edge is adjacent to a triangle if it is a part of the border of that triangle.

In contrary to an interior edge, a boundary edge is adjacent to exactly one triangle. A boundary vertex is a vertex used to define a boundary edge. Thus, a closed cycle of boundary edges defines a hole. A singular vertex has more than two adjacent boundary edges. A non-manifold edge has more than two adjacent triangles. A

manifold mesh has no non-manifold edges and no singular vertices, but may have boundaries.

In this work, we assume that all the meshes are triangular, oriented, manifold and connected. In particular, two separate holes will have no vertices in common (otherwise those would be singular), and a given hole will not have islands (otherwise the mesh would not be connected). The original mesh and the mesh that fills the hole are respectively called the surrounding mesh and the inner mesh. These hypothesizes differ from those of [15] who is able to build a watertight triangle mesh from any initial models represented as a polygon soup.

4 Identification and cleaning of the hole contour

Given the previous hypothesizes, the hole contour can be automatically identified while looking for a closed cycle of boundary edges. A single contour has been identified and coloured in blue on the example of figure 3.
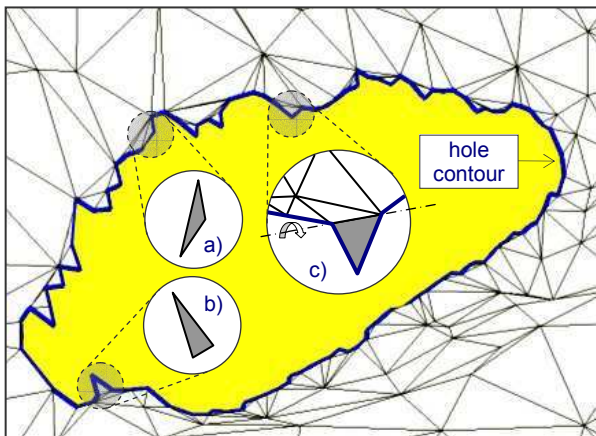

Figure 3: Hole contour identification and undesired triangles resulting from the scanner noise (a,b,c).

The scanner noise may cause the appearance of degenerated triangles along the hole contour. These thin triangles can be of two types depending on the size of their angles: either one or two small angles (figures 3.a and 3.b). But the scanner noise may also give some triangles of the hole boundary widely varying orientations with respect to the surrounding mesh (fig. 3.c). If nothing is done, all these triangles may lead to undesired undulations of the inner mesh that will fill in the hole. To overcome these drawbacks, it is mandatory to clean the contour of the hole before inserting a new topological mesh. To this aim, three algorithms have been designed and are presented hereafter. Naturally, they are optional and can be runned or not. Their use may slightly affect the shape of the surrounding mesh triangles very close

to the hole contour. But it does not matter since the affected triangles are in an uncertainty area.

4.1 Deletion of badly oriented triangles

A badly oriented triangle has an orientation very different from the orientation of its adjacent triangles. In order to identify such a triangle, one could control the angles between the triangle and its adjacents triangles. If one of the angles is superior to a given threshold, the triangle is treated. But this algorithm requires the specification of a threshold and the computation of the angles.

One can also notice that a triangle defined by two boundary edges can rotate around its interior edge without affecting the other triangles (fig. 3.c). This degree of freedom, combined with the uncertainty on the position of the vertex connected to the two boundary edges (scanner noise), may lead to a bad orientation of the triangle with respect to its adjacent triangles. To prevent such configurations, we delete recursively all the triangles defined by two boundary edges (black triangles on the example of figure 4.a).
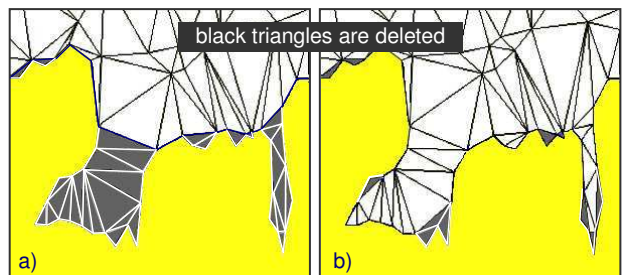

Figure 4: Deletion of potentially badly oriented triangles which preserves (b), or not (a), some significant details of the contour.

Using this process some significant details of the contour may disappear. To overcome this limit, an additional condition should be satisfied at each step of the recursive deletion process: a triangle having two boundary edges can be deleted only if its adjacent triangle does not have any boundary edge (fig. 4.b).

4.2 Treatment of degenerated triangles

Degenerated triangles are characterized by exactly one or two small angles (figures 3.a and 3.b). To identify such triangles without computing their three angles, we use an indicator first introduced to check the quality of Finite Element meshes [6]. The *aspect ratio Q* of a given triangle is defined as follows:

$$Q = \alpha \frac{\rho}{h} = \alpha \frac{S}{hp} \qquad (1)$$

3

with $\alpha = 2\sqrt{3}$ a normalization coefficient so that $Q = 1$ for an equilateral triangle, h is the longest edge length, ρ is the radius of the incircle, S is the area of the triangle and p its half-perimeter. This quality factor belongs to the interval $[0, 1]$. The limit zero corresponds to flat triangles. It is commonly accepted that a triangulation is a good one, with respect to the Finite Element approximation, if the aspect ratio of the worst triangle is greater than 0.5. In our approach, we are less demanding than for a Finite Element analysis and we accept more degenerated triangles. As a consequence, only the triangles whose ratio is smaller than 0.25 will be considered as degenerated. This threshold has been found empirically while analyzing the evolution of the ratio according to the possible evolution of the triangle angles.
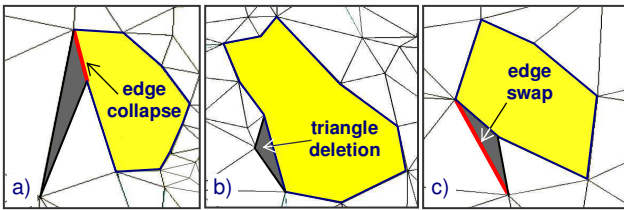

Figure 5: Three types of degenerated triangles.

All the triangles considered as degenerated are then treated separately according to three possible configurations. First, if the triangle has one small edge with respect to the two others, the smallest edge is collapsed (fig. 5.a). Otherwise, if the longest edge is a boundary edge, the triangle is deleted (fig. 5.b), else the longest edge is swapped (fig. 5.c).

4.3 Homogenization of the edge lengths

Depending on the way the object is scanned, e.g. multiple positions and resolutions of the scanner, the lengths of the boundary edges may vary in a quite significant way. Consequently, the surrounding triangles can be of heterogenous sizes. This can be wished or not. In case such a behaviour is undesired, our homogenization algorithm is applied.

To this aim, the average length ℓ_a of the contour is first computed. It corresponds to the sum of the boundary edge lengths ℓ_i divided by the number of edges forming the initial contour. Each edge i of the oriented contour is then tested separately: if the ratio $\ell_i/\ell_a > 1.5$, the edge is split in two; if the ratio $\ell_i/\ell_a < 0.7$, the edge is collapsed to one of its two extremities; otherwise the edge is not modified. At each step of this loop, some triangles can be either deleted or added to reveal the topological modifications.

The thresholds used for the tests have been determined empirically. They should be optimized in a future version of our homogenization algorithm. Figure 6.b shows the result of this algorithm applied on the example of figure 6.a.
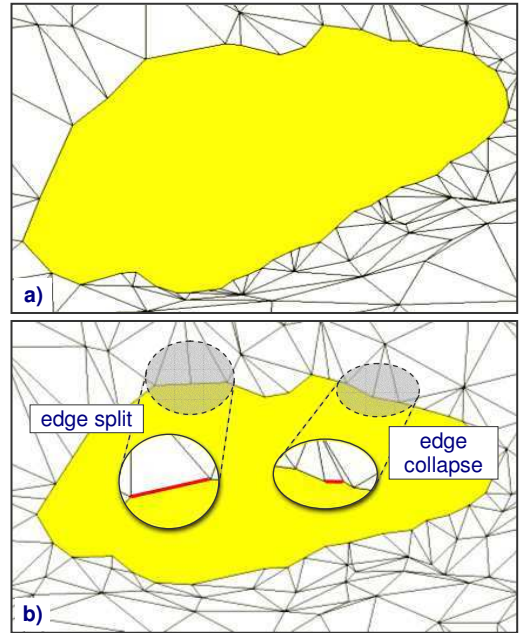

Figure 6: Treatment of degenerated and badly oriented triangles (a) and homogenization of the edge lengths (b) on the example of figure 3.

5 Topological filling

The contour being cleaned, the topological filling process can start. In this first version of our approach, the hole is filled in with a topological grid coming from the meshing of an unit 2D disk having the same number of boundary vertices as the hole contour (figures 7.a and 7.b). The connections between the inner vertices, spread arbitrarily on the disk, are defined using the Delaunay triangulation. The grid contour is then stretched and merged with the hole contour. The new position of the inner vertices is unknown and will be computed in the next step, i.e. the deformation of the inner mesh (section 6). They can be gathered together at the barycentre of the boundary vertices (fig. 7.c) since their initial position does not affect the result of the deformation.

Using this topological grid is a way to decouple the topology from the geometry: the hole is first filled and the shape of the inner mesh is then determined. It could easily be extended to the treatment of holes having several islands. This way of determining the topology of the inner mesh gives good results when the hole to be filled looks like a disk. Unfortunately, it may lead to degenerated triangles when the hole does not look like a disk or when the boundary edges are of heterogeneous sizes. To overcome these limits, the hole triangulation method of

Liepa [19] could be used, his umbrella fairing operator being replaced by our new minimization that uses an analogy with a mechanical model of bar network to simulate the minimization of the curvature variation between the surrounding and inner meshes. This is possible because the proposed overall process is modular (fig. 2) which enables the changing of a module, the topological filling in the present case, by another. We come back on these aspects in section 7.
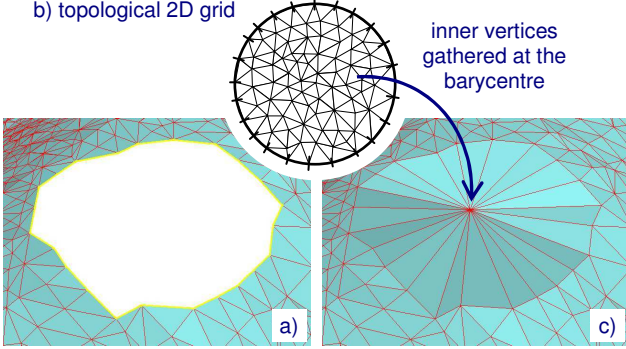

Figure 7: Topological filling using an unit disk.

6 Mesh Deformation

Once the hole is filled with a topological grid, the position of the free vertices, i.e. the inner mesh vertices that are not on the contour, have to be determined. In order to choose one among the whole set of possible resulting shapes, an additional criterion has to be specified. For example, either the inner mesh may simply fill the hole and minimize its area (fig. 2.c), or the inner mesh may fill the hole while satisfying tangency/curvature blending conditions with the surrounding mesh (fig. 2.d). In our approach, the position of the free vertices results from the resolution of a minimization problem based on a mechanical model of bar networks coupled to the whole mesh (inner + surrounding meshes). In case several inner meshes have to be shaped, several bar networks are coupled to the various pieces and the position of all the free vertices are computed in a single deformation step. This is one of the strengths of our approach.

6.1 Mechanical model of bar networks

The first idea to alter the shape of a given network is to modify one by one the position of its free vertices. Figure 8.a shows how a network can be modified through the displacement of a single vertex numbered 2. These basic manipulations are time consuming since they require the manipulation of all the vertices before achieving the desired shape.

To overcome these limits, we use a mechanical model of bar network coupled to the geometric model [27]. In this way, the vertices of the geometric model and the connections between them match respectively the nodes and the bars of our bar network. Each bar can be seen as a spring with a null initial length and a stiffness q_i (more precisely a force density). To preserve the static equilibrium state of the structure, external forces f_i have to be applied to the nodes (fig. 8.b). If these external forces were not applied to the network, all the nodes would be gathered together at a single point. The linear relationships between the external forces and the position of the nodes enable intuitive shape modifications through the manipulation of a restricted set of external forces. On the example of figure 8.c, solely the external force applied to the second node is pertubed to produce the displacement of all the free nodes.
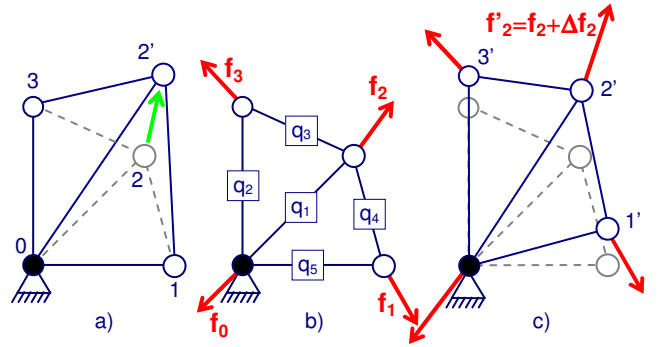

Figure 8: Deformation of a network with (c) and without (a) a coupling to our mechanical model (b).

Formalization:

Given $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$, the three vectors containing the components of the 3D coordinates of the N_n nodes of the bar network coupled to the control vertices of the meshes (a single bar network is built on the surrounding and inner meshes), the $\boldsymbol{f}_x$, $\boldsymbol{f}_y$ and $\boldsymbol{f}_z$ components of the external forces applied to these nodes can be obtained by using the following $(3 \times N_n)$ equations expressing the bar network static equilibrium:

$$\begin{aligned} \boldsymbol{f}_x &= ({}^t\mathbf{C}.\mathbf{Q}.\mathbf{C}).\boldsymbol{x}, \\ \boldsymbol{f}_y &= ({}^t\mathbf{C}.\mathbf{Q}.\mathbf{C}).\boldsymbol{y}, \\ \boldsymbol{f}_z &= ({}^t\mathbf{C}.\mathbf{Q}.\mathbf{C}).\boldsymbol{z}, \end{aligned} \qquad (2)$$

where $\mathbf{Q}$ is the force density matrix of size $(N_b \times N_b)$ being N_b the number of bars. $Q_{ij} = q_j.\delta_{ij}$ with δ_{ij} the Kroenecker's symbol and $q_j = f_j/\ell_j$ the force density into the jth bar of length ℓ_j. $\mathbf{C}$ is the branch-node matrix of size $(N_b \times N_n)$ expressing the connectivity of the network [23].

Following [25], a distinction between free and blocked nodes can be performed in the equations (2). It gives rise

to the two following sets of equations:

$$\begin{aligned}
\boldsymbol{f}_{fn_x} &= \mathbf{D}_f.\boldsymbol{x}_{fn} + \mathbf{D}_{bf}.\boldsymbol{x}_{bn}, \\
\boldsymbol{f}_{fn_y} &= \mathbf{D}_f.\boldsymbol{y}_{fn} + \mathbf{D}_{bf}.\boldsymbol{y}_{bn}, \\
\boldsymbol{f}_{fn_z} &= \mathbf{D}_f.\boldsymbol{z}_{fn} + \mathbf{D}_{bf}.\boldsymbol{z}_{bn},
\end{aligned} \tag{3}$$

for the external forces applied to free nodes and:

$$\begin{aligned}
\boldsymbol{f}_{bn_x} &= {}^t\mathbf{D}_{bf}.\boldsymbol{x}_{fn} + \mathbf{D}_b.\boldsymbol{x}_{bn}, \\
\boldsymbol{f}_{bn_y} &= {}^t\mathbf{D}_{bf}.\boldsymbol{y}_{fn} + \mathbf{D}_b.\boldsymbol{y}_{bn}, \\
\boldsymbol{f}_{bn_z} &= {}^t\mathbf{D}_{bf}.\boldsymbol{z}_{fn} + \mathbf{D}_b.\boldsymbol{z}_{bn},
\end{aligned} \tag{4}$$

for the external forces applied at blocked nodes. The different $\mathbf{D}_i$ matrices can be easily obtained through the decompositions of the ${}^t\mathbf{C}.\mathbf{Q}.\mathbf{C}$ matrix.

Conversely, being given a set of external forces applied to the nodes of the bar network, the position of the free nodes are given by:

$$\begin{aligned}
\boldsymbol{x}_{fn} &= (\mathbf{D}_f)^{-1}.(\boldsymbol{f}_{fn_x} - \mathbf{D}_{bf}.\boldsymbol{x}_{bn}), \\
\boldsymbol{y}_{fn} &= (\mathbf{D}_f)^{-1}.(\boldsymbol{f}_{fn_y} - \mathbf{D}_{bf}.\boldsymbol{y}_{bn}), \\
\boldsymbol{z}_{fn} &= (\mathbf{D}_f)^{-1}.(\boldsymbol{f}_{fn_z} - \mathbf{D}_{bf}.\boldsymbol{z}_{bn}).
\end{aligned} \tag{5}$$

These last equations show how it is possible to manipulate indirectly the node positions through the manipulation of external forces (see [23] for the treatment of configurations where the $\mathbf{D}_f$ matrix is singular). The deformation process unknowns are not anymore the free nodes positions but the external forces applied to the free nodes. However, some external forces applied to blocked nodes connected to at least one free bar, i.e. a bar that changes of length during the process, may vary during the deformation and can therefore take part to the definition of the objective function to be minimized (section 6.3). They are linearly dependent on the external forces applied at free nodes and the corresponding equations can be obtained using a combination of equations (4) and (5):

$$\begin{aligned}
\boldsymbol{f}_{bn_x} &= {}^t\mathbf{D}_{bf}.(\mathbf{D}_f)^{-1}.(\boldsymbol{f}_{fn_x} - \mathbf{D}_{bf}.\boldsymbol{x}_{bn}) + \mathbf{D}_b.\boldsymbol{x}_{bn}, \\
\boldsymbol{f}_{bn_y} &= {}^t\mathbf{D}_{bf}.(\mathbf{D}_f)^{-1}.(\boldsymbol{f}_{fn_y} - \mathbf{D}_{bf}.\boldsymbol{y}_{bn}) + \mathbf{D}_b.\boldsymbol{y}_{bn}, \\
\boldsymbol{f}_{bn_z} &= {}^t\mathbf{D}_{bf}.(\mathbf{D}_f)^{-1}.(\boldsymbol{f}_{fn_z} - \mathbf{D}_{bf}.\boldsymbol{z}_{bn}) + \mathbf{D}_b.\boldsymbol{z}_{bn}.
\end{aligned} \tag{6}$$

Finally, this mechanical model of bar networks is independent on the type of underlying geometry. It has been used by [25] for the shape tuning of Fully Free Form Deformation Features [24]. Several networks were simultaneously coupled to the control polygon/poly-hedron of NURBS curves and surfaces. Here, the way to new perspectives is opened while enabling the coupling of the bar networks to the vertices of a mesh that has to be manipulated.

6.2   Optimization problem formulation

To enable the deformation of the inner mesh, a single bar network is coupled to the whole mesh (inner + surrounding meshes) and all the nodes coupled to the vertices of

the surrounding mesh are blocked. The external forces applied to the free nodes are so many unknowns the user still has to adjust to find the appropriate position of the free nodes. This is not acceptable. In fact, these manipulations must be transparent for a neophyte who may be interested in the specification of meaningful geometric constraints and shape behaviours. This corresponds to the formulation of an optimization problem:

$$\left\{ \begin{array}{l} \boldsymbol{G}(\boldsymbol{F}) = \boldsymbol{0} \\ \min \phi(\boldsymbol{F}) \end{array} \right. \tag{7}$$

where the unknown vector $\boldsymbol{F}$ contains the components of all the external forces applied to free nodes. Such a formulation clearly shows the decoupling that exists between:

- the **geometric constraints** that may be imposed either between the inner and surrounding meshes (e.g. position, tangency or curvature constraints) or directly on the inner mesh to further specify its interior shape. These constraints produce a set of possibly non-linear equations linking directly the position of the free vertices. The resulting constraint vector $\boldsymbol{G}$ can be expressed as a function of the external forces applied to the free nodes using the equation (5).

- and the **objective function** ϕ to minimize. This is a higher level parameter enabling the specification of various deformation behaviours through the combination of several geometric and/or mechanical quantities relative to the bar network. Using such control parameters is a way to achieve semantic-based manipulations which fits well the current preoccupations of the researchers [1]. It is a key point of our deformation method since we are not restricted to the classical strain energy minimization.

In our approach, no constraints are specified between the inner and surrounding meshes. The inner mesh deformation is obtained by minimization of an objective function that takes into account the geometric information (e.g. normal or curvature evolution) available on the surrounding mesh (section 6.3). Thus, we avoid non-linear equations and over-constrained configurations. Despite everything, the use of additional constraints to adjust the shape of the inner mesh is discussed in section 6.4. This additonal possibility is also one of the strengths of our approach.

6.3   Curvature variation minimization

In this section, the optimization problem is reduced to the minimization of the objective function $\phi(\boldsymbol{F})$. Any combination of basic quantities that can be expressed as a quadratic form of the unknown vector can be used to
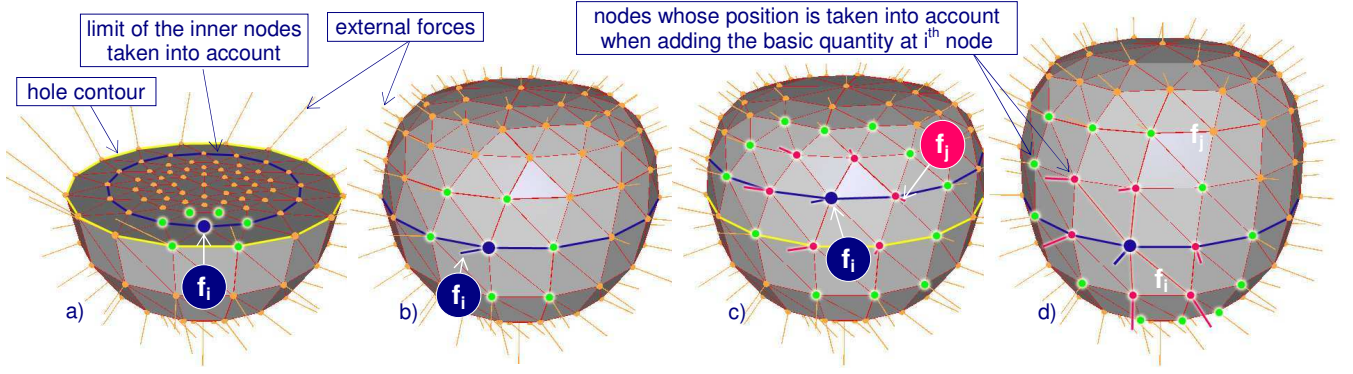
Figure 9: Minimization of ϕ_F applied to free nodes and to nodes of the hole contour (a,b). Minimization of $\phi_{\Delta^r F}$ applied to free nodes and to nodes of the hole contour (c,d). Vertices influencing the external force $\boldsymbol{f}_i$ are tagged.

define this single objective function [25]. The minimization of these quantities gives rise to a system of linear equations that can be solve directly without requiring an iterative process.
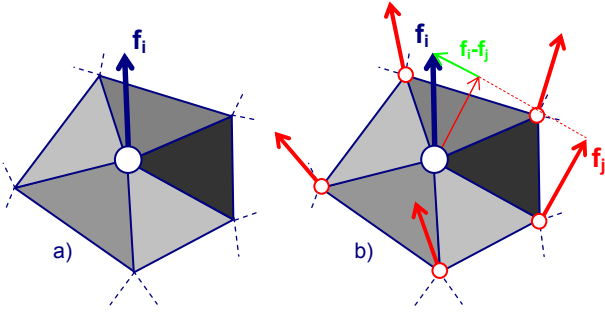

Figure 10: External forces taken into account when minimizing either the external force applied to the ith node (a) or the variations of this external force with respect to the surrounding ones (b).

Among the various quantities that may be used in the present context, there is the minimization of the external forces:

$$\phi_F = \sum_i \boldsymbol{f}_i^2 = \sum_i f_{x_i}^2 + f_{y_i}^2 + f_{z_i}^2 \qquad (8)$$

The external forces applied to the free nodes being independent the ones from the others (fig. 10.a), the minimization of this sum on all the free nodes will always produce a null unknown vector (no crossed terms in the quadratic form). This will tend to minimize the curvature of the underlying geometry while producing areas as planar as possible (fig. 9.a). This can be interesting in the context of tensile textile structures [33]. Moreover, one can notice that an external force is only influenced both by the position of the node on which it is applied and by the position of the vertices directly connected to it (fig. 9.a). As a consequence, solely the information relative to the position of the vertices of the hole contour are considered when minimizing the external forces

applied to the free nodes. This does not answer our requirements since neither the tangency evolution nor the curvature evolution are preserved across the hole contour. To take into account more geometric information on the surrounding mesh, one can add to the sum the external forces applied to the hole contour nodes. Using the equations (6), one can obtain a quadratic objective function with squared and crossed terms. The influence area is larger since it uses geometric information relative to the first ring of the surrounding mesh, i.e. relative to all the vertices that are connected to the hole contour vertices (fig. 9.b). Unfortunately, such a minimization cannot pretend more than the preservation of the tangency with the surrounding mesh.

To rich the level of curvature preservation, one should use geometric information relative to the second ring of the surrounding mesh. Starting from the simple remark that the variation between two successive external forces simulates in some sense the evolution of the curvature between the two nodes, the minimization of the following quantity is proposed:

$$\phi_{\Delta^r F} = \sum_i \sum_j \left[\boldsymbol{f}_i - \boldsymbol{f}_j \right]^2 \qquad (9)$$

It enables the minimization of the variations between couples of external forces applied to nodes connected together. The influence area of these basic quantities is greater than in the previous case (fig. 9.c). This is due both to the fact that all the external forces applied to the nodes j, directly connected to a given node i, are also taken into account (fig. 10.b) and to the fact that each external force is influenced both by the position of the node on which it is applied and by the position of the vertices directly connected to it. But our requirements are still not answered when the minimization of this sum is performed on all the free nodes (fig. 9.c). As previously, solely the first ring of the surrounding mesh is used. In order to take into account the second ring, the basic quantities relative to the nodes of the hole contour
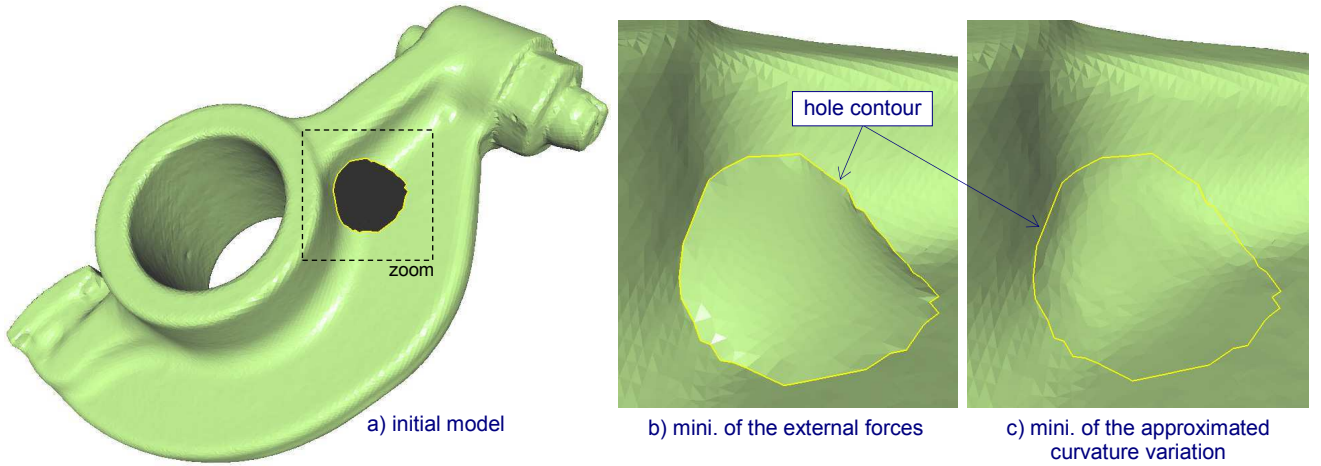
Figure 11: Filling a hole inside the polyhedral model of a digitalized rocker arm (courtesy of Cyberware). Once the hole is filled with a topological grid, the position of the inner vertices is obtained by either the minimization of the external forces (b) or the minimization of the external forces relative variation (b).

also have to be used (fig. 9.d). In this case, the evolution of the curvature tends to be preserved when crossing the hole contour. In fact, the curvature variation is simulated by the external forces relative variation. Here, one can notice that the two first rings of the inner mesh seem to be obtained by symmetry of the two first rings of the surrounding mesh according to the plan containing the hole contour.

To finish, it is not interesting to use more rings of the surrounding mesh since the supplementary terms would be constant and would not affect the result of the minimization. In addition, being able to minimize the curvature variation between the inner and surrounding meshes is enough for most of the mechanical engineering applications.

6.4 Shape adjustments

The strength of our approach lies in the use of a deformation technique to determine the shape of the inner mesh. Not only it is possible to use an adapted minimization criterion $\phi(\boldsymbol{F})$ but the specification of an optional constraint vector $\boldsymbol{G}(\boldsymbol{F})$ may also be envisaged. This is very interesting when the user holds extra information concerning the shape of the missing part. For example, these information can be extracted from a photography of the object.

Technically, any constraint expressed as a function of the coordinates of the mesh vertices is acceptable. Using the equations (5), one will always be able to come back to a set of equations that depend on the unknown vector $\boldsymbol{F}$. In case of non-linear constraints, a linearization is performed at the first order and the resolution using a Lagrangian becomes iterative. Among the various possibilities, [23] proposes a set of point constraints linking

a parametric point lying on a mesh (not necessarily one of the vertices) with a geometric point in 3D space. Either position, tangency or distance conditions may be imposed between the two. An approach similar to the one of [5] could also be used to insert planar area inside the inner mesh. In case of a watertight mesh, one could also constrain the interior volume as described in [18].

However, due to the non-linearity of some of these constraints, the iterative resolution process may not converge easily. Moreover, if too many constraints are specified, or if the constraints are badly specified, the system may become over-constrained. These two arguments further justify our choice of a new minimization to smooth the inner mesh, the use of geometric constraints being relegated to the rank of additional tools to shape the interior of the inner mesh (section 6.2). An example is proposed in section 7.

7 Results and discussion

The first example concerns the polyhedral model of a rocker arm (courtesy of Cyberware). To illustrate the power of the proposed approach, a hole has been created in a complex blending area (fig. 11.a). After a cleaning operation, the hole is filled in with a topological grid having 50 boundary vertices. The 3D position of the 190 free inner vertices can be computed while solving a (570×570) linear system coming from the minimization of the external forces (fig. 11.b). No additional constraints are specified. Using this objective function, nothing ensure a smooth connection between the inner and surrounding meshes and the repaired area is easily identifiable on the final polyhedral model. These limits are overcome while using the minimization of the exter-
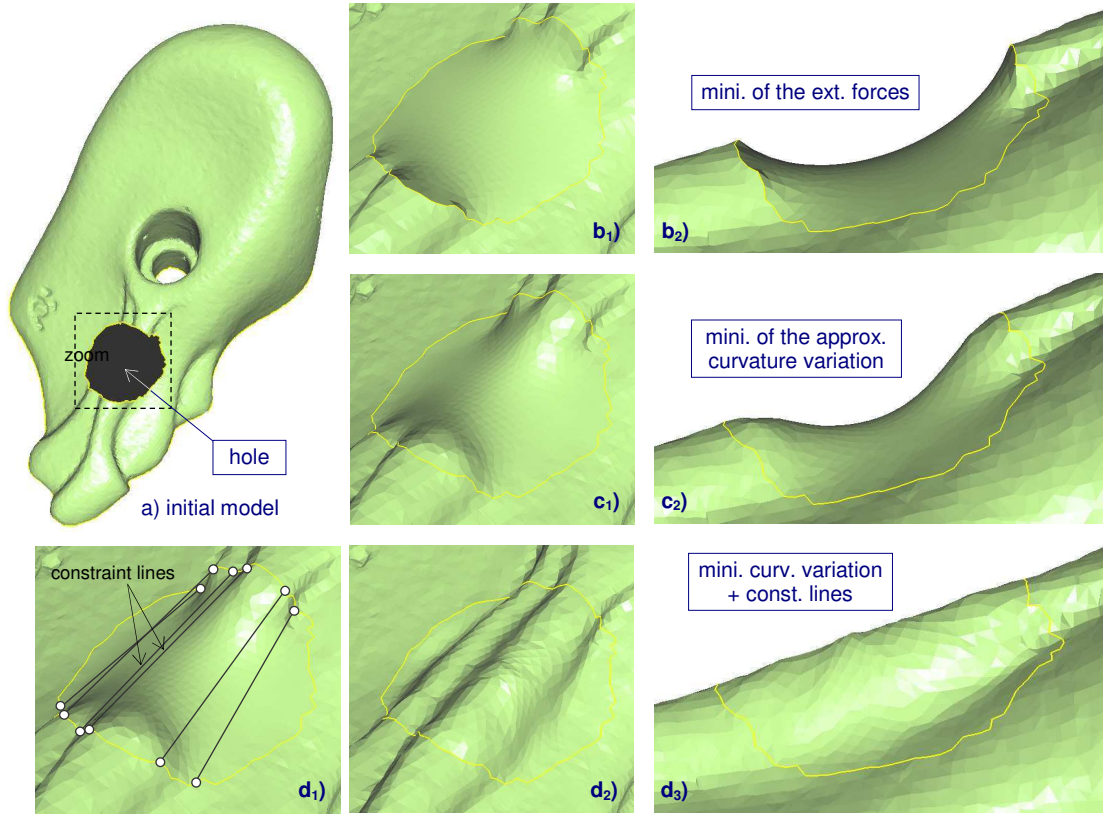
Figure 12: Filling a hole inside the polyhedral model of a digitalized artificial climbing hold (courtesy Tomoadour). The minimization of the external forces does not produce a smooth connection with the surrounding mesh (b_i). This is improved while using the new minimization (c_i). Details of the inner shape can be retrieved while specifying manually additional constraint lines the polyhedron has to satisfy (d_i).

nal forces relative variation. In this case, the sum in the equation (9) is performed on all the free and boundary vertices. Thus, the inserted area is difficultly distinguishable from the initial polyhedron (fig. 11.c).

The second example concerns the polyhedral model of an artificial climbing hold. Here again, a hole is created inside a complex area. A topological grid with 100 boundary vertices and about 700 inner vertices is used to fill in the hole (fig. 12.a). The shape of the inner mesh can be determined while using either the external forces or the curvature variation minimizations (figures b_i and c_i). In both cases, no additional constraints are specified. Even if the new minimization produces a smooth connection with the surrounding mesh, the inserted area is easily distinguishable. This example shows the limit of the proposed approach. Effectively, as it is, our approach is unable to retrieve complex missing shapes and the stiffener that was on the initial model has disappeared. To overcome this limit, some shape adjustments can be performed while specifying additional constraints (section 6.4). Here, the user has created manually six constraint lines joining six couples of points. The various points have been chosen according to the shape of the surrounding stiffener, i.e. in such a way that the

constraint lines prolong what could be the edges of the stiffener (fig. 12.d_1). These lines are then discretized into a set of point constraints constraining parametric points lying on the mesh with geometric points lying on the curves. The points on the mesh are obtained by projection. These constraints plus the minimization of the approximated curvature variation form an optimization problem whose solution is close to the shape that was initially removed (figures 12.d_2 and d_3).

Finally, the figure 13 shows a comparison between our approach and the fairing technique of Liepa that uses a second-order umbrella operator [19]. The minimization of the external forces applied to the free nodes produces a shape that simply connects with the surrounding mesh while minimizing its area (fig. 13.a). The second-order umbrella operator of Liepa is applied to the free nodes. As a consequence, it takes into account solely the first ring of vertices of the surrounding mesh. The two meshes can only connect while satisfying tangency blending conditions (fig. 13.b). To further improve the quality of the connection, the minimization of the external forces relative variation is applied to the free and boundary vertices. It takes into account the two first rings of vertices of the surrounding mesh. Thus, the two meshes connect

9

smoothly while minimizing the curvature variation between the two (fig. 13.c). On this example, one can notice that the inflection in the center of the inner mesh is more pronounced when using the external forces relative variation minimization than in the other cases. In fact, the area of the inner mesh affected by the shape of the surrounding mesh is larger than in the other cases. Consequently, the inner mesh has to evolve faster in the less affected areas.
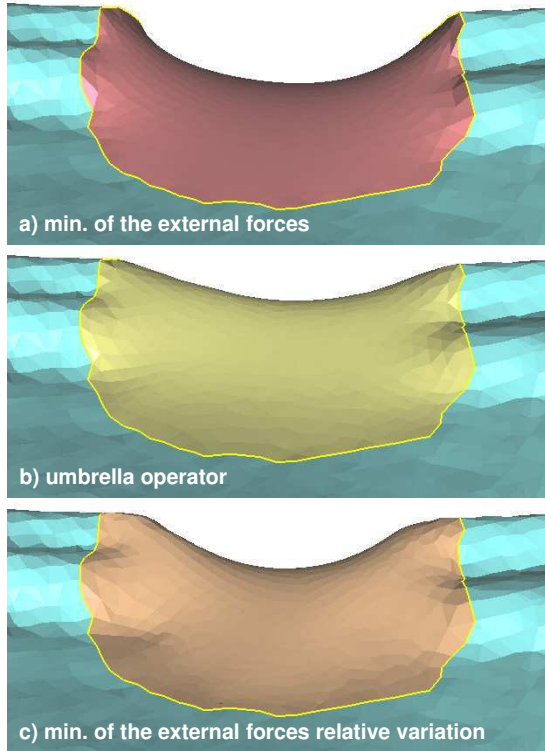

Figure 13: Comparison between the fairing approaches on the example of the artificial climbing hold.

8   Conclusion

In this paper, we propose a complete process to fill in holes in meshes. It goes from the detection and cleaning of the hole contour to the deformation of a topological grid built from the boundary vertices. New methods and tools have been imagined at each of these steps. The deformation itself results from the resolution of an optimization problem where the constraints are optionally specified to shape the interior of the filled area. A new type of objective function is notably proposed and enables the minimization of an approximation of the variation of the curvature between the inner and surrounding meshes. More precisely, the curvature variation is simulated by the variation between external forces applied to the nodes of a linear mechanical model coupled to the meshes. Thus, the equations system to be solved is linear. The whole process has been validated on several examples coming from our prototype software. It gives very good results in configurations where the holes look like a disk. A comparison with the second-ordeer umbrella operator of Liepa is proposed and clearly demonstrates that the proposed minimization enables a smoother connection with the surrounding mesh. Anyhow, additional works should be performed to be able to adapt both the topology (disk, ring and so on) and the shape (circle, rectangle and so on) of the grid to the shape of the hole. At the end, these improvements of the topological filling step should enable the filling of any holes that can be found during the reverse engineering of products.

References

[1] Aim@Shape. Advanced and innovative models and tools for the development of semantic-based systems for handling, acquiring, and processing knowledge embedded in multidimensional digital objects. *European Network of Excellence, Key Action: 2.3.1.7 Semantic-based knowledge systems, VI Framework, URL: http://www.aimatshape.net.*

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on computer graphics and interactive techniques*, 1998.

[3] G. Barequet and M. Sharir. Filling gaps in the boundary of a polyhedron. *Computer Aided Geometric Design*, 12(2):207–229, 1995.

[4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[5] V. Cheutet, J-P. Pernot, J-C. Léon, B. Falcidieno, and F. Giannini. Insertion of planar areas into free-form surfaces in early product design. In *Proceedings of ASME DAC'05: International Design Engineering Technical Conferences And Design Automation Conference*, 2005.

[6] P.G. Ciarlet. *The finite element method for elliptic problem.* North Holland, 1978.

[7] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A finite element method for surface restoration with smooth boundary conditions. *Computer Aided Geometric Design*, 21(5):427–445, 2004.

[8] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Processing of SIGGRAPH'96*, pages 303–312, 1996.

[9] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Processing of the first International Symposium on 3D Data, Visualization and Transmission*, 2002.

[10] M. Desbrun, M. Meyer, P. Schrder, and A.H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1999.

[11] M. Eck and H. Hoppe. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[12] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1), 1994.

[13] D.R. Forsey and R.H. Bartels. Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, 14(2), 1995.

[14] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface Reconstruction from Unorganized Points. In *Proceedings of the 19th annual conference on computer graphics*, pages 71–78, 1992.

[15] Tao Ju. Robust repair of polygonal models. *ACM Transaction on Graphics (TOG)*, 23(3):888–895, 2004.

[16] L. Kobbelt, S. Campagna, J. Vorsatz, and H-P. Seidel. Interactive Multi-Resolution Modeling on Arbitrary Meshes. In *Proceedings of SIGGRAPH'98*, pages 105–114, 1998.

[17] A. Lee, H. Moreton, and H. Hoppe. Displaced Subdivision Surfaces. In *Proceedings SIGGRAPH'00*, 2000.

[18] L. Lien and J. Kajiya. A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra. *IEEE Comput. Graph. Appl.*, 4(9):35–41, 1984.

[19] P. Liepa. Filling Holes in Meshes. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry Processing (SGP'03)*, pages 200–205, 2003.

[20] W. Ma and J.P. Kruth. Parameterization of randomly measured points for least squares fitting of b-spline curves and surfaces. *Computer-Aided Design*, 27(9):663–675, 1995.

[21] W. Ma and N. Zhao. Catmull-Clark surface fitting for reverse engineering applications. In *Proceedings of Geometric Modeling and Processing, Theory and Applications*, pages 274–283, 2000.

[22] F.S. Nooruddin and G. Turk. Simplification and Repair of Polygonal Models Using Volumetric Techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):191–205, 2003.

[23] J-P. Pernot. Fully Free Form Deformation Features for Aesthetic and Engineering Designs. *Ph.D. thesis, INP-Grenoble (France) and University of Genoa (Italy). http://www.3s.hmg.inpg.fr/ci/doc/*, 2004.

[24] J-P. Pernot, B. Falcidieno, F. Giannini, and J-C. Léon. Fully Free Form Deformation Features for aesthetic shapes design. *Journal of Engineering Design (JED)*, 16(2):115–133, 2005.

[25] J-P. Pernot, S. Guillet, J-C. Léon, B. Falcidieno, and F. Giannini. Shape Tuning in Fully Free Form Deformation Features. *Journal of Computing and Information Science in Engineering (JCISE)*, 5(1):95–103, 2005.

[26] R. Pfeifle and H-P. Seidel. Triangular B-Splines for Blending and Filling of Polygonal Holes. In *Proceedings of Graphics Interface*, pages 186–193, 1996.

[27] H.J. Schek. The force density method for form finding and computation of general networks. *Computational Methods in Applied Mechanics and Engineering*, 3(2):115–134, 1974.

[28] R. Schneider and L. Kobbelt. Geometric fairing of irregular meshes for free-form surface design. *Computer-Aided Geometric Design*, 18(4):359–379, 2001.

[29] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Geometry Processing*, pages 179–188, 2004.

[30] S. Takeuchi, T. Kanai, H. Suzuki, K. Shimada, and F. Kimura. Subdivision surface fitting with QEM-based mesh simplification and reconstruction of approximated B-spline surfaces. In *Proceedings of the 8th pacific conference on Computer Graphics and Applications*, pages 202–212, 2000.

[31] L.S. Tekumalla and E. Cohen. A hole-filling algorithm for triangular meshes. *Technical report UUCS-04-019, School of Computing, University of Utah, USA, http://www.cs.utah.edu/techreports/*, 2004.

[32] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro. Inpainting surface holes. In *International Conference on Image Processing (ICIP)*, 2003.

[33] P. Véron, P. Trompette, and J-C. Léon. Integrated design and collaborative engineering of fabric structures. *Engineering with Computers*, 14:23–35, 1998.

[34] J. Wang and M.M. Oliveira. A hole-filling strategy for reconstruction of smooth surfaces in range images. In *Proceedings of SIBGRAPI'03*, 2003.