Towards recovery of complex shapes in meshes using digital images for reverse engineering applications

Minica Panchetti, Jean-Philippe Pernot*, Philippe Véron

LSIS - UMR CNRS 6168, Arts et Métiers ParisTech, 2 cours des Arts et Métiers, 13617 Aix-en-Provence Cedex 1, France

Keywords:
Reverse engineering
Hole-filling algorithm
Mesh deformation
Shape From Shading
Curvature variation minimization

A B S T R A C T

When an object owns complex shapes, or when its outer surfaces are simply inaccessible, some of its parts may not be captured during its reverse engineering. These deficiencies in the point cloud result in a set of holes in the reconstructed mesh. This paper deals with the use of information extracted from digital images to recover missing areas of a physical object. The proposed algorithm fills in these holes by solving an optimization problem that combines two kinds of information: (1) the geometric information available on the surrounding of the holes, (2) the information contained in an image of the real object. The constraints come from the image irradiance equation, a first-order non-linear partial differential equation that links the position of the mesh vertices to the light intensity of the image pixels. The blending conditions are satisfied by using an objective function based on a mechanical model of bar network that simulates the curvature evolution over the mesh. The inherent shortcomings both to the current hole-filling algorithms and the resolution of the image irradiance equations are overcome.

1. Introduction

Nowadays, three-dimensional geometric models have become common in mechanical engineering. They are the intermediate representations shared between the actors of the design process. They also support the semantic information [1] relative to the different steps (e.g. simulation parameters, materials, manufacturing entities). To be able to analyse and/or to update an existing product whose digital model is unavailable, it is mandatory to first reconstruct it with specific Reverse Engineering (RE) techniques. The idea consists in creating a digital representation of an existing physical object by interpolating or approximating a point cloud acquired either with a laser scanner or a Coordinate Measuring Machine. The result can either be a triangle mesh [2–5] or a set of B-Spline/NURBS surfaces [6–8] or a subdivision surface [9–11]. Most of the time, the triangulation of the point cloud is a preliminary step to the creation of more complex surfaces. Hence, it is crucial to be able to produce a triangle mesh that best fits the object's outer surface.

Unfortunately, depending on: (1) the type of the data acquisition system (2) the adopted methodology (e.g. single or multiple view points) (3) the shape complexity, the resulting point cloud may be deficient in some inaccessible areas. This is due to the optical occlusion phenomenon that is illustrated by the example on Fig. 1. The areas of the scanned column that are visible from the view point (c) were invisible from the acquisition view point (b). Thus, several undesired holes have appeared in the triangulation.

It seems obvious that the presence of these holes is not acceptable. Such a lack of information may lead to unexpected results when performing numerical simulations, e.g. finite element simulation, clash detection, or simply when reconstructing a digital representation of a monument. Therefore, new methods have to be proposed to fill in these holes. To enable an efficient recovery of the missing parts, these algorithms should fulfil several criteria that have been illustrated by the example in Fig. 2. Here, a hole has been included voluntarily in a complex area of an artificial climbing hold (Fig. 2(a)). The criteria are relative to:

- *the shape of the newly inserted triangles* that should be as similar as possible to the shape of the triangles surrounding the holes. The resulting patching meshes of Fig. 2(b_1) and (b_2) do not fulfil this requirement since the inserted triangles have either too different sizes or privileged directions. If we suppose that the mechanism/algorithm used to obtain the final shape is not sensitive to the topology of the patching mesh, these two first criteria can be achieved by using a post-processing step. They would smooth and refine the resulting mesh so that the inserted triangles' size and density match those of the surrounding triangles. For instance, a metric such as the one proposed by [12] which is a quality factor, can be used to measure the shape of the inserted triangles. It is an indicator usually adopted to check the quality of Finite Element meshes. It identifies "degenerated" triangles, i.e. triangles defined by one or two small angles,

* Corresponding author. Tel.: +33 4 42 93 81 85; fax: +33 4 42 93 81 00.
E-mail addresses: jean-philippe.pernot@ensam.fr, jean-philippe.pernot@aix.ensam.fr (J.-P. Pernot).

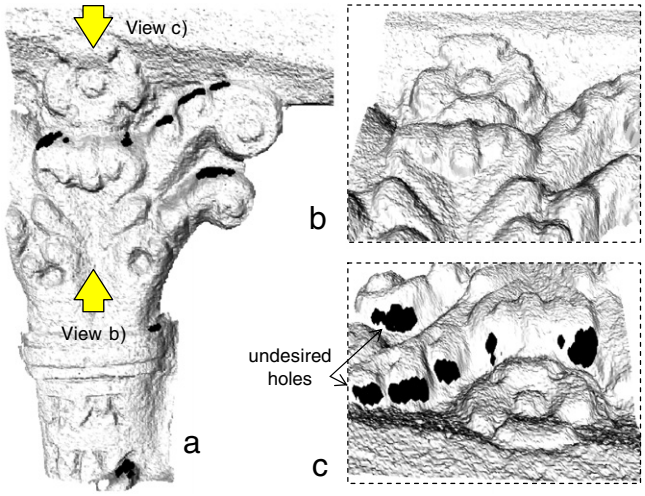


Fig. 1. Column of an edifice (a) scanned from the single view point (b). Undesired holes appeared (c) in areas which are invisible from the view point (b).

from "equilateral" triangles. The resolution of the inserted mesh should fit the surrounding mesh by refinement method such as the one proposed by [13] (see Section 3).

- *the blending continuity conditions* between the inserted and surrounding meshes. To fill in holes in complex free-form shapes, it is mandatory to produce patches that are visually similar to the surrounding meshes. To this end, some position (Fig. 2(c_1)), tangency, or curvature (Fig. 2(c_2)) blending conditions should be accessible by the user.
- *the quality of the shape recovery*. The shapes of the patching meshes have to follow as much as possible the evolution of the shapes of the physical object that is reverse engineered. It is certainly the most difficult criterion to achieve. Fig. 2(d) shows the part of the stiffener that has been removed and that should potentially be recovered by a hole-filling algorithm.

To the best of our knowledge, no existing algorithm is able to perform the ideal recovery illustrated in Fig. 2(d). In this paper, we overcome these limitations and propose a new approach to recover complex shapes in meshes. The idea is to extract the missing data while using 2D images that have been potentially taken during the acquisition step. These data form a set of non-linear constraints that are used as an input to our mesh deformation process. Its behaviour tends to minimize the curvature variation between the inserted and surrounding meshes. More precisely, an optimization problem is solved, which is made of: (1) an objective function to be minimized, based on a mechanical model of bar network that simulates the curvature evolution over the mesh (2) constraints, coming from the image irradiance equation. This equation is a first-order non-linear partial differential equation linking the position of the mesh vertices to the light intensity of the image pixels.

As shown in Fig. 3, the proposed approach becomes highly valuable when some areas of the object to be reverse engineered (amphitheatre of the "Art et Métiers ParisTech—Aix-en-Provence" in the present case) cannot be reached by the laser scanner. Here, the digital model of the amphitheatre results from the triangulation of three scans obtained with a laser scanner located at three distinct positions in front of the edifice. There are more than one million triangles. Using such a methodology, some areas of the monument are invisible from the floor thus resulting in a set of undesired holes (Fig. 3(a)). To circumvent these shortcomings, it seems promising to complete the definition of the digital model while using images taken from the sky (Fig. 3(b_1) and (b_2)) with a camera attached to a radio-commanded helicopter (Fig. 3(c_1) and (c_2)). More generally, when acquiring tall edifices, the use of a laser scanner together with a camera taking pictures from the sky can be really helpful and less expensive than moving the scanner to unreachable positions, even more true when considering huge laser scanning sessions. For example, the digitalization of the "Château Comtal" in the city of Carcassonne (France) has led up to more than 45 millions of 3D points and 540 photographs representing 60 GB of data (courtesy MAP GAMSAU laboratory [14]). In that specific case, the photographs are

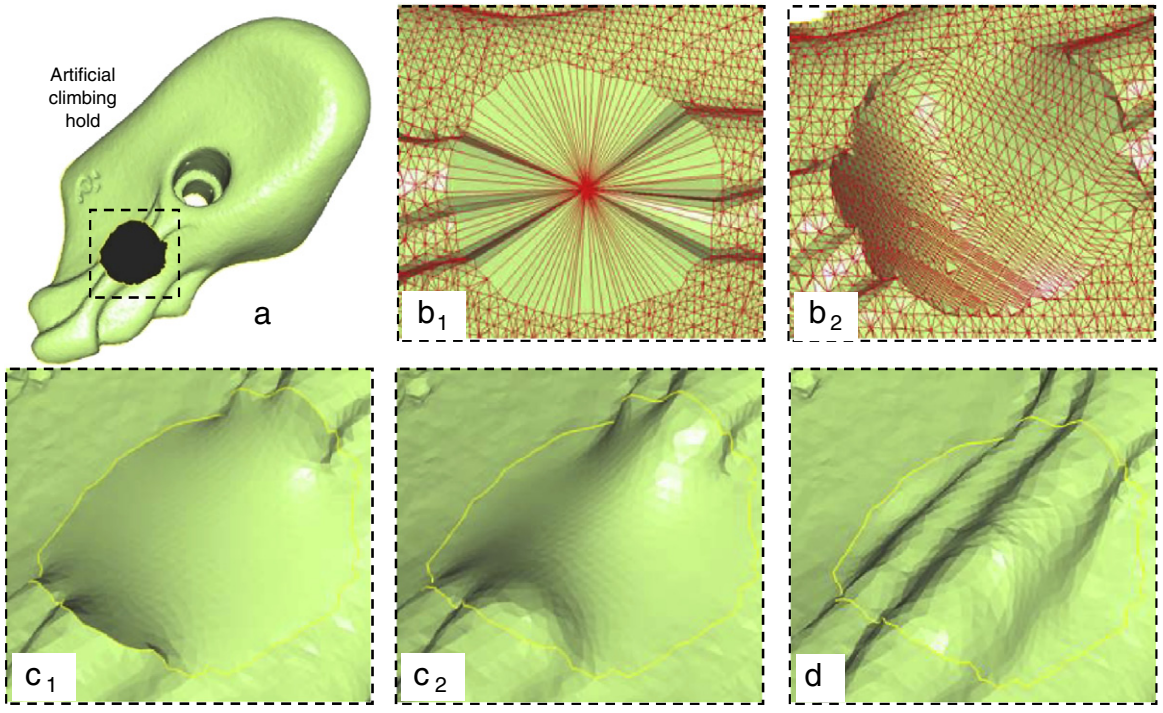


Fig. 2. Various criteria to fulfil when filling a hole inside the mesh of an artificial climbing hold (a, courtesy Tomoadour): Criteria relative to the triangles size (b_1), the preferred directions of the inserted mesh (b_2), the type of blending condition between the inserted and surrounding meshes (c_1, c_2) or the quality of the recovery (d).

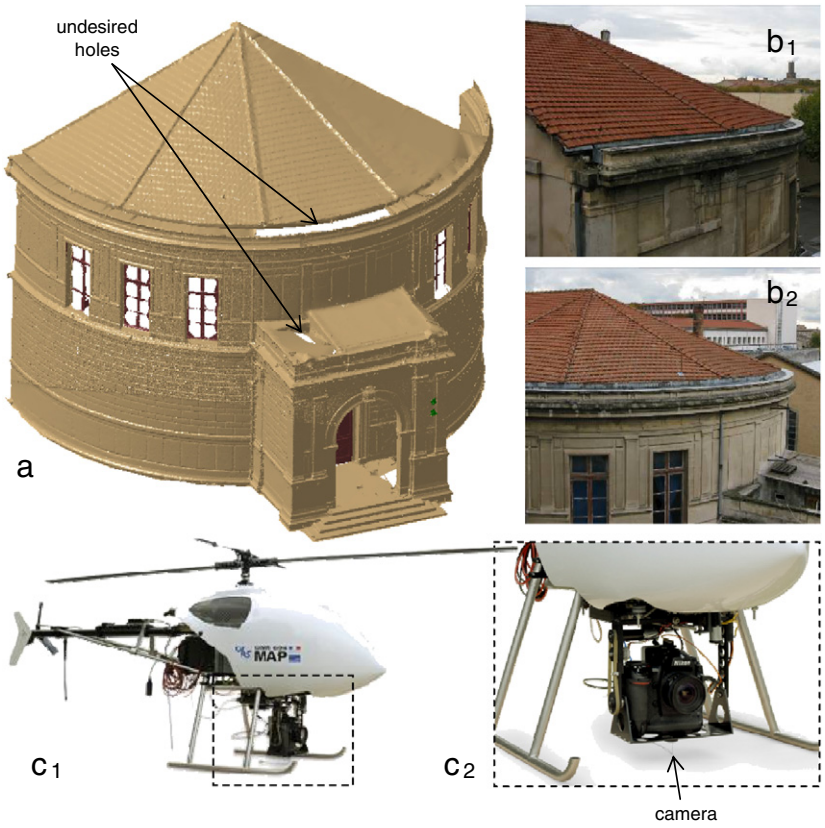


Fig. 3. Scanned amphitheatre of the Arts et Métiers ParisTech—Aix-en-Provence (a) revealing holes to be filled in using photographs (b_1, b_2) taken from the sky by a radio-commanded helicopter (c_1, c_2), courtesy GAMSAU laboratory.

calibrated, which means that the position and the orientation of the camera which took them are known. Finally, it is clear that the method is not interesting to fill in holes whose missing shape is not complex. In the actual implementation, the degree of complexity of the holes is decided by the user. Complex holes are filled in using the whole process whereas simple holes are filled in using solely the minimization of the curvature variation without any additional constraints extracted from any pictures. Thus, for simple identified holes, the criterion relative to the quality of the shape recovery is not considered. Of course, our approach cannot recover shapes from areas inaccessible to both the scanner and the camera.

This paper is written as follows. Section 2 discusses on the different attempts for developing some hole-filling algorithms. It comes out that none of the existing methods are able to answer the three previously introduced quality criteria. Section 3 presents our developed method which combines the best from both SFS technologies and mesh deformation techniques. Section 4 introduces the Shape From Shading technique that is used to constrain the inserted mesh through the image irradiance non-linear equation. The mesh deformation technique that is used to position automatically the mesh vertices according to intuitive deformation behaviours is then presented in Section 5.1. It uses a linear mechanical model that ensures a smooth blending between the inserted and surrounding meshes. Section 5 addresses the way the resulting optimization problem (set of non-linear equations plus minimization based on a mechanical model) is solved. Section 6 discusses the results. The last section concludes this work and introduces research perspectives.

2. Related work

The state-of-the-art can be decomposed into two main parts. The first part gathers together the methods that can be used to fill in holes in meshes by satisfying the two first previously introduced criteria, i.e. the definition of triangles with appropriate shapes and sizes according to the surrounding mesh and, the definition of a smooth blending between the inserted and surrounding meshes. The second part gathers together the techniques that are used to reconstruct 3D models from 2D images. Actually, our newly developed algorithm combines some of these approaches to fill in holes while satisfying the three criteria (Fig. 2).

2.1. Hole-filling algorithms

Several methods have been proposed to fill in holes in meshes. They can be decomposed into two main categories: the geometric and non-geometric approaches.

Among the non-geometric approaches, Curless et al. [15] use a volumetric representation to detect the areas that have to be filled in. Davis et al. [16] 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. A similar approach has been developed by Nooruddin et al. [17] and Ju [18] for the simplification and the repairing of polygonal meshes. Verdera et al. [19] 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 on the holes and their neighbourhood. Because these equations are anisotropic and based on geometry, they lead to a slightly slower algorithm than the one of Davis et al. [16]. Clarenz et al. [20] 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

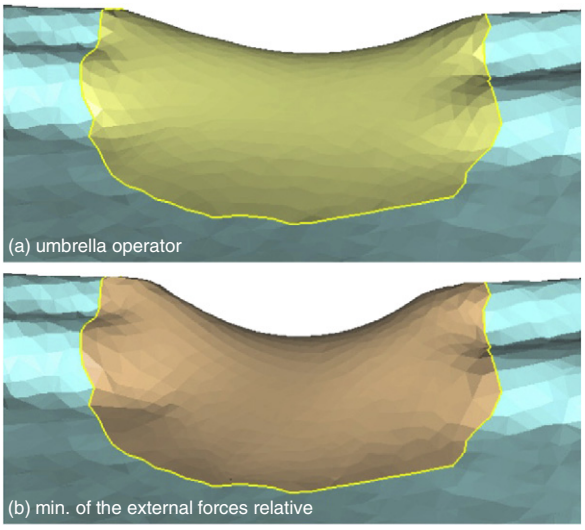


Fig. 4. Comparison of blending results obtained with the method of Liepa [22] (a) and our approach [30] (b) on the example of the artificial climbing hold (courtesy Tomoadour).

continuity with the surrounding mesh. A similar approach using non-linear partial differential equations to solve surface modeling problems has also been proposed by Xu et al. [21].

Considering the geometric approaches, Liepa [22] proposes a sequential filling process. The hole is first detected and filled with a minimum area triangulation of its 3D contour according to the method of Barequet et al. [23]. The triangulation is then refined so that the triangle density agrees with the density of the triangles of the surrounding mesh as developed by Pfeifle et al. [24]. The filled hole is finally smooth with a fairing technique based on an umbrella operator proposed by Kobbelt et al. [25]. The faired meshed is obtained by solving a system of linear equations. It satisfies tangency blending conditions with the initial mesh. Schneider et al. [26] 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 Desbrun et al. [27] requires an iterative resolution process. Other approaches are based on the Moving Least Squares projection which induces a non-linear minimization process [28,29].

Among all these approaches, very few are able to fill in holes while satisfying curvature blending conditions (G^2 continuity). Actually, those that are capable to do it often require the resolution of non-linear equations which may slow down the overall process. In [31], we have proposed a non-geometric approach enabling the filling of holes while approximating the minimization of the curvature evolution through the holes boundary. Unfortunately, this approach was unable to recover complex shapes as the one presented in Fig. 2(d). But this was predictable. Here, there exists an infinite number of solutions that satisfy curvature blending conditions with the surrounding mesh. In the Fig. 4, the blending obtained with our approach is compared to the one obtained by the fairing technique of Liepa [22] that uses a second-order umbrella operator. The umbrella operator solely achieve tangency blending conditions (Fig. 4(a)) whereas our method tends to minimize the curvature variation across the hole boundary (Fig. 4(b)). Actually, this is due to the fact that the umbrella operator takes into account solely a first ring of vertices of the surrounding mesh whereas our method uses the two first rings of vertices. These notions are further detailed in Section 5.2.

Similarly to the previously mentioned methods, the approach developed in [31] was unable to recover complex shapes as the one presented on Fig. 3(d). But this was predictable since there exists an infinite number of solutions that satisfy curvature blending conditions with the surrounding mesh. The result can either be more or less flat, or it can undulate a lot, or it can be whatever while always satisfying the same blending conditions. Thus, to overcome these limitations, the missing information have to be found elsewhere. This is why we thought to use images that can be easily taken during the acquisition step.

2.2. Shape From Shading approaches

Many techniques aim at recovery depth information from one or several images. They are called Shape From *X* methods, where *X* can be Stereo, Texture, Shadows, Contours, Shading. Among them, the Shape From Stereo techniques use a pair of photographs of the same 3D surface but taken from two distinct view points to obtain the 3D information by triangulation [32]. Shape From Texture methods only reach indications about the 3D surface and require the texture to be homogeneous [33]. Shape From Shadows works analyse the contour formation of a known object (called the screen object) on the surface to recover [34]. Shape From Contours techniques convert contours of a rotated object profiles into a volumetric description [35]. Finally, solely the Shape From Shading methods aim at recovering the 3D information from only one grey level image, which is our goal.

Shape From Shading (SFS) approaches have originated during the 70s. They aim at recovering 3D models from one or more 2D images. For example, starting from the 2D image of Fig. 5(a), it is possible to build the 3D mesh of Fig. 5(b). Horn first attempted to write the SFS problem [36] as a problem of resolution of the brightness equation (see Section 4 for more details on the equation), a non-linear first-order Partial Differential Equation (PDE). Thereafter, several methods have been developed to try to solve this equation. However, the developed methods suffer from poor quality results, and the convergence is not ensured. In addition, it has been proved that the solution of the Shape From Shading problem is not unique [37–39]. The main difficulty remains the existence of the concave/convex ambiguities that is illustrated in Fig. 6. Effectively, depending on its orientation, a 2D image can lead to the definition of either a concave or a convex shape. Finally, the resulting mesh owns privileged Cartesian directions (Fig. 5(c)), corresponding to the orientation of the pixels in the 2D image, which does not fit the requirements introduced in Section 1.

Unfortunately, all these methods reconstruct meshes starting from scratch, i.e. without taking into account the information available on an initial mesh. In the hole-filling context, some information are already available on the surrounding mesh and the images should solely be used to complete this information. The approach developed by Turk et al. [43] also uses a collection of range images to reconstruct a model starting from scratch. Then, the resulting meshes are merged and smoothed while computing a mean surface. This does not fit our requirements in terms of preservation of the information surrounding the hole.

2.3. Combining SFS constraints with geometric constraints

To the best of our knowledge, no research has been undertaken to try to fill in holes in meshes using SFS constraints. Despite everything, the approach of [44] is interesting since it combines SFS constraints with geometric constraints to reconstruct 3D shapes. The image information is akin to stereo and SFS, while the geometric information may be provided in the form of 3D points or features, or 2D silhouettes. The use of geometric constraints, to further constrain the shape of the reconstructed mesh, helps the resolution of the convex/concave ambiguity. The solution is obtained by minimizing an energy function defined from both the SFS and

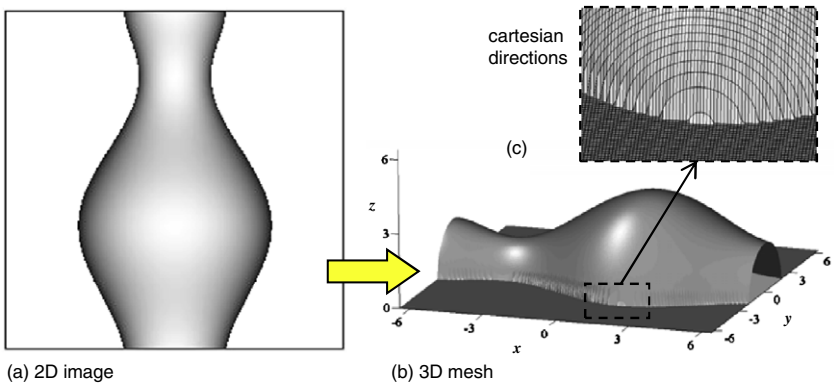
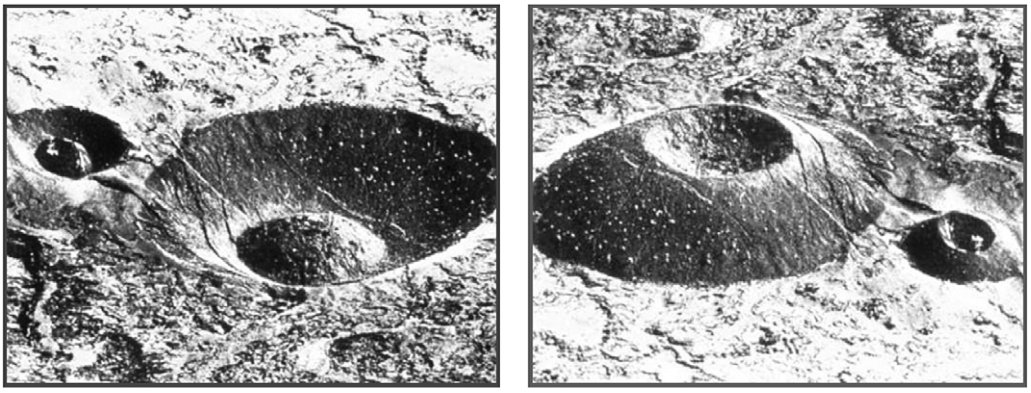


Fig. 5. From 2D image (a) to 3D mesh (b) using Shape From Shading [40,41]. The resulting mesh owns Cartesian directions (c).



(a) Initial image. (b) After a rotation of 180°.

Fig. 6. An example of the concave/convex ambiguity [42]. Depending on its orientation, an image can reveal shapes that look like craters (a) or volcanoes (b).

geometric constraints. The total energy is a sum of terms whose magnitudes depend either on the image or geometric information and are therefore not necessarily commensurate. Thus, the definition of weights between the different magnitudes is difficult. Moreover, solely the specification of position geometric constraints is available which does not fit our requirements in terms of curvature blending conditions between the inserted and surrounding meshes (Fig. 2(c_2)).

3. Unified framework to fill in holes in meshes using SFS constraints

In the proposed approach, the advantages of both SFS technologies and mesh deformation techniques are exploited at best within a unified framework. Thus, the disadvantages of the SFS methods are annihilated by the advantages of the deformation techniques, and vice versa. The combination of these two techniques is performed through the resolution of an optimization problem so that:

- *the unknowns* are the 3D positions of the vertices of an initial coarse triangulation that does not privilege any Cartesian direction, and so that the sizes of the inserted triangles evolve according to the sizes of the surrounding triangles (criterion 1 introduced in Section 1). However, and since the proposed approach is modular, the use of constraints such as those described in [43] can be considered to deform the inserted coarse patch and to get a directly fitted mesh.
- *the constraints* come from the Shape From Shading equations which reflect the way the missing area behaves (criterion 3). The partial differential equations are approximated by using finite differences. It results in a set of non-linear equations (see Section 4).
- *the functional to be minimized* is a quadratic form of the unknowns that uses a linear mechanical model of bar networks to simulate the minimization of the curvature evolution over the mesh (see Section 5.1). Thus, we achieve curvature blending conditions between the inserted and surrounding meshes (criterion 2).

To be able to set up this optimization problem, several steps have to be followed sequentially (Fig. 7):

(1) the hole contour is identified by looking for boundary edges, i.e. those edges sharing a unique face. Following [31], the contour is then cleaned to remove badly oriented and degenerated triangles (b). These irrelevant triangles are due to the scanner noise and must not be considered as safe data. The pretreatments concern the thin triangles, i.e. those having one or two small angles (Fig. 8(a) and (b)), and the badly oriented ones, i.e. those having widely varying orientations with respect to the surrounding mesh (Fig. 8(c)). In this work, all the meshes are supposed to be triangular, oriented, 2-manifold with boundary 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). Similarly, soups of triangles are not considered here.
(2) the hole is then triangulated and refined (c) using the approach developed in [13]. Here, the triangle mesh does not own privileged directions, which is a significant difference with the other SFS approaches developed in the literature. Moreover, the refinement is performed so that the evolution of the inserted triangles size matches the evolution of the surrounding triangles' size.

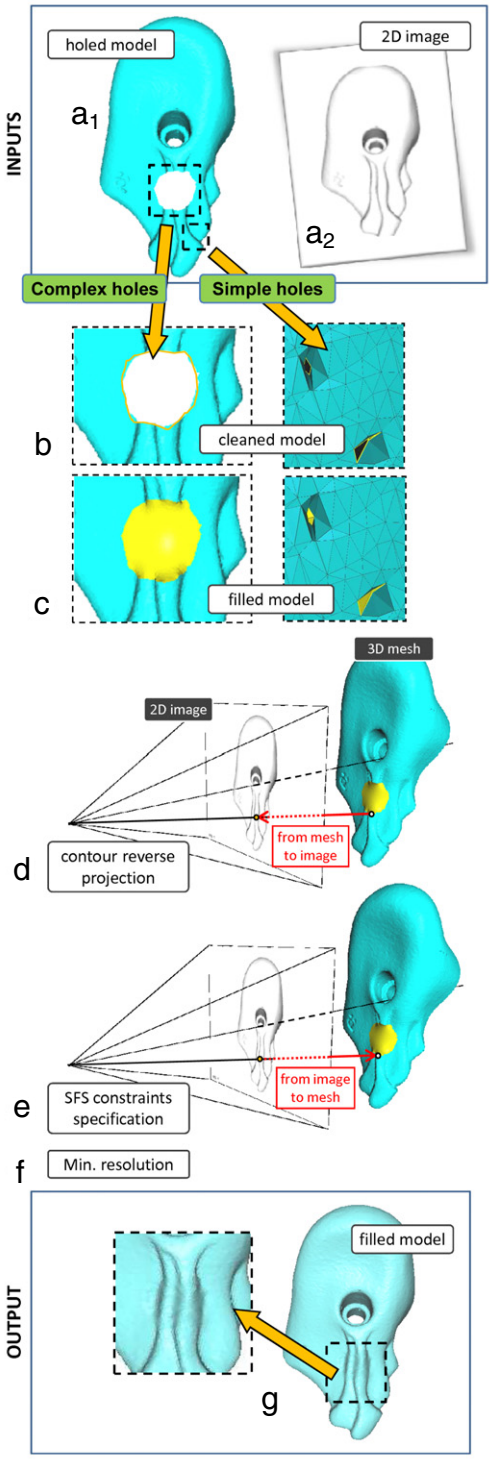


Fig. 7. Overall filling process on the holed climbing hold (a_1): After the identification and cleaning of their contours (b), the simple and complex holes are successively triangulated and refined (c). User-identified complex holes are then reverse-projected in the 2D image (d) to be able to specify the SFS constraints (e) before deforming the patching meshes according to these constraints and curvature blending conditions (f).

(3) if the image was not already calibrated, the position, orientation and focal distance of the camera which has taken the image are then computed from a set of user-specified couples pixel-vertex. These parameters are thus expressed within the global reference frame where the mesh lies. The hole contour is then reverse-projected inside the calibrated image to identify the pixels that correspond to the missing area (d).

(4) a restricted set of the identified pixels is then used to set up the SFS constraints that locally constrain the shape of the mesh (e).

(5) the mechanical model of bar network is then built and coupled to the inserted mesh vertices as well as to a set of triangles in the close neighbourhood of the hole contour (f). The coupling to these outer vertices ensures the definition of an objective function which does not only depend on the unknown positions, but also on the position of the vertices located in the surrounding mesh which is crucial to get curvature blending conditions.

(6) the optimization problem is then solved using the Differential Evolution method adapted to non-linear cases. Finally, the constraints are satisfied and the curvature variation is minimized over the entire mesh (g).

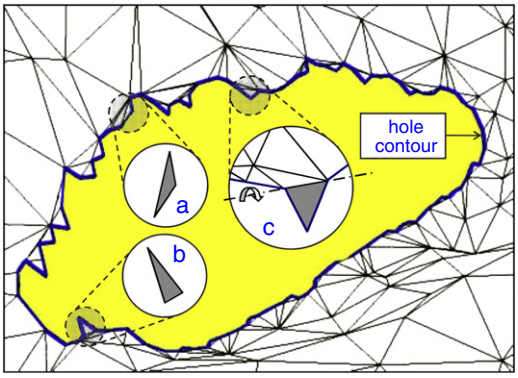


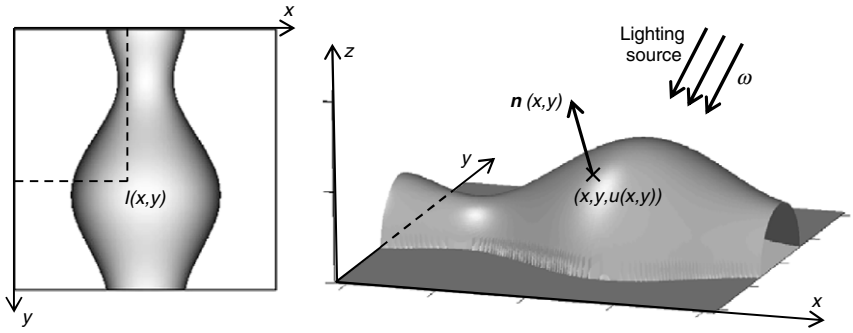
Fig. 8. Hole contour identification and undesired triangles resulting from the scanner noise [45].

Table 1
Time spent percentages during each step of the overall process.

	Step (b)	Step (c)	Step (d)	Step (e)	Step (f)
Simple holes	25%/20%[a]	75%/60%[a]	0%/0%[a]	0%/0%[a]	0%/20%[a]
Complex holes	1%	3%	5%	5%	86%

[a] Corresponds to the case where no SFS constraints are applied.

Excepting Steps (1) and (2), whose references are given and which are not indispensable for the understanding of the proposed approach, all the steps are detailed hereafter. On the example of Fig. 7, one can notice that simple and complex holes have been distinguished. A simple hole is characterized by the fact that its missing shape is not complex. Since the automatic evaluation of the complexity of the missing shape is not yet available, this step is left to the user who has to decide whether a hole has to follow the whole process or not, including the use of the information contained in the images or not. Such a distinguishing speeds up the overall filling process. This is visible on the percentages of the time spent in the different steps (Table 1). Thus, when treating complex holes, most of the time is spent to specify the SFS constraints and solve the optimization problem. When treating simple holes, if the optional minimization of the curvature variation is not performed, the most expensive step is the triangulation. Finally, it is important to notice that our modular process can be parallelized since each hole can be filled in independently of the other ones. Here, the time spent to calibrate the images has not been taken into account. In the actual implementation, this step is manual and time consuming since it requires the identification of several couples of pixel-vertex.



(a) 2D image. (b) Bi-parametric surface.

Fig. 9. Image (a) of a surface (b) lighted with a light located at the infinity.

4. Constraining the inserted mesh using SFS equations in orthogonal projection

Basically, the SFS technique constrains a bi-parametric surface $z = u(x, y)$ to satisfy the image irradiance equation, i.e. the equation characterizing the way the light reflects on the surface. In this section, we describe how these classical SFS constraints are adapted to constrain the shape of the inserted mesh.

4.1. SFS basic assumptions and Eikonal equation

Given $z = u(x, y)$, with $(x, y) \in \Omega \subset \mathbb{R}^2$, the bi-parametric surface lighted with a lighting source located at the infinity and having $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$ as unit direction vector (Fig. 9(b)), the *image irradiance equation* is written as follows [46,41]:

$$R(\boldsymbol{n}(x, y)) = I(x, y) \quad \forall (x, y) \in \Omega \tag{1}$$

with I the brightness function supposed to be known through the analysis of the image (Fig. 9(a)), and R the reflectance function that gives the value of the light reflection at the 3D point $(x, y, u(x, y))$. $\boldsymbol{n}(x, y)$ is the unit normal to the surface so that:

$$\boldsymbol{n}(x, y) = \frac{1}{\sqrt{1 + p^2 + q^2}}(-p, -q, 1) \quad \forall (x, y) \in \Omega \tag{2}$$

with $p = \frac{\partial u}{\partial x}(x, y)$ and $q = \frac{\partial u}{\partial y}(x, y)$ the components of $\nabla u(x, y) = (p, q)$.

If we assume that the surface is Lambertian, i.e. the reflectance function is directly proportional to the cosine of the angle between the source light direction and the surface normal (Fig. 9(b)), which is mostly the case when dealing with real objects, one can add the following equation:

$$R(\boldsymbol{n}(x, y)) = \boldsymbol{\omega} \cdot \boldsymbol{n}(x, y) \quad \forall (x, y) \in \Omega. \tag{3}$$

Thus, Eq. (1) becomes the following first-order non-linear partial differential equation:

$$I(x, y)\sqrt{1 + \|\nabla u(x, y)\|^2} + (\omega_1, \omega_2) \cdot \nabla u(x, y) - \omega_3 = 0 \quad \forall (x, y) \in \Omega. \tag{4}$$

Setting $\boldsymbol{\omega} = (0, 0, 1)$, Eq. (4) becomes the Eikonal equation:

$$\left[\frac{\partial u}{\partial x}(x, y)\right]^2 + \left[\frac{\partial u}{\partial y}(x, y)\right]^2 = \frac{1}{I(x, y)^2} - 1 \quad \forall (x, y) \in \Omega \tag{5}$$

with the boundary conditions $u(x, y) = 0 \quad \forall (x, y) \in \partial\Omega$.

Conversely, given an RGB image, the intensity function $I(x, y)$ can be computed according to the recommendation of the International Commission on Illumination. The solution of the Eikonal equation (5) can then be computed to find the resulting surface $u(x, y)$. Technically, and since the image contains a finite set of pixels (x_i, y_j), the resolution of this equation produces a set of 3D points $(x_i, y_j, u(x_i, y_j))$ forming a discrete surface oriented along the parametric direction x and y. Thus, if this method was directly applied to our hole-filling context, the resulting inserted mesh would own privileged directions. This does not satisfy the criterion 1 introduced in Section 1. This limit is overcome in the next subsections.

Moreover, to the best of our knowledge, solely simple boundary conditions such as $u(x, y) = g(x, y)\ \forall (x, y) \in \partial\Omega$ have been tried in the SFS context. But either tangency or curvature boundary conditions along the boundary $\partial\Omega$ would be mandatory in our hole-filling context. However, in this paper, we did not try to use directly such complex constraints on $u(x, y)$. Instead, we use a linear mechanical model of bar network to simulate curvature blending conditions (Section 5.1).

Coming back to the adopted shading model, one can notice that real surfaces are not always Lambertian, that the position of the light source is often unknown and that there often exist several light sources. However, since our process is modular, these restrictive hypotheses can still be reconsidered in the future. For example, one could try to implement more sophisticated and realistic lighting models such as the one of Bakshi et al. [47]. The work of Wang et al. [28] may also be interesting to recover the position and the orientation of the light source.

4.2. Camera calibration and reverse projection of the hole contour

To be able to constrain the three-dimensional mesh defined in its global reference frame, with information extracted from an image defined in its own two-dimensional space, a camera calibration step is mandatory. This step aims at retrieving the fundamental projection matrix $\mathbf{M}$ linking the coordinates (X_{ij}, Y_{ij}, Z_{ij}) of a 3D point $\mathbf{P}_{ij}$ to the coordinates (x_i, y_j) of the corresponding point $\mathbf{p}_{ij}$ in the 2D image (Fig. 10). This matrix can either be known when the location and characteristics of the camera that took the image are recorded by a GPS, or it can be computed *a posteriori* while selecting pairs of $\mathbf{P}_{ij}$ and $\mathbf{p}_{ij}$ points [48,49], or using autocalibration methods [50]. It is known when the image corresponds to a screenshot:

$$\begin{bmatrix} x_i \\ y_j \\ 1 \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \\ 1 \end{bmatrix}. \tag{6}$$

Once the image is calibrated and the matrix is known, the reverse projection of the hole contour into the 2D image can be performed. In this paper, we assume that the reverse projection defines a closed polyline which does not self-intersect. Following [51], the set of pixels contained in the polygon are extracted. They are used hereafter to constrain the mesh deformation.

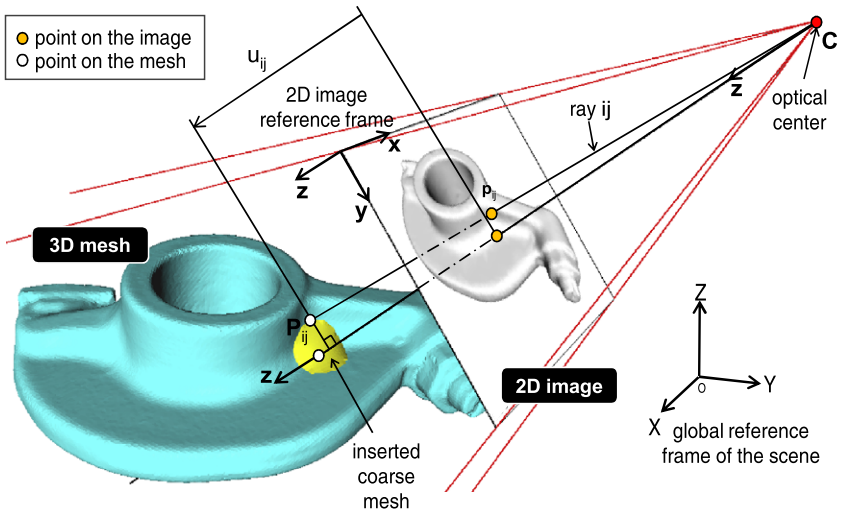


Fig. 10. Linking a 3D mesh to a 2D image through the camera calibration.

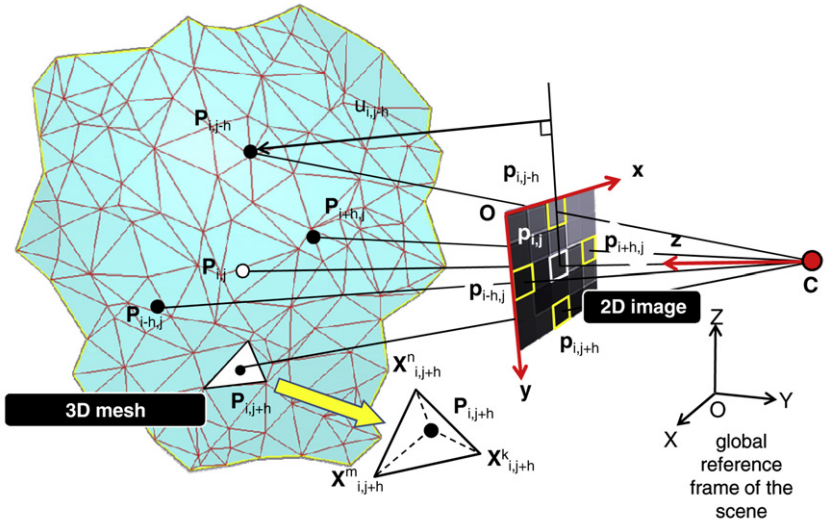


Fig. 11. Correspondence between a pixel $\boldsymbol{p}_{ij}$ and a point $\boldsymbol{P}_{ij}$ located in a triangle of the inserted mesh.

4.3. Specification of a SFS constraint between a pixel and the 3D mesh

Given a pixel $\mathbf{p}_{ij}$ of the 2D image, whose brightness $I(x_i, y_j)$ is known, we adapt the Eikonal equation (5) so that it constrains the mesh around the corresponding 3D point $\mathbf{P}_{ij}$ (see Fig. 11 for the notations of this section). To approximate the partial derivatives, we use a second-order centred finite difference scheme. Thus, we obtain:

$$\left[\frac{u_{i+h,j} - u_{i-h,j}}{\Delta x_{ij,h}}\right]^2 + \left[\frac{u_{i,j+h} - u_{i,j-h}}{\Delta y_{ij,h}}\right]^2 = \frac{1}{I(x_i, y_j)^2} - 1 \tag{7}$$

where h is an integer representing the step of the grid that is used in the finite difference scheme. Actually, it is used to better distribute the constraints over the mesh and specify the extent of the constraint influence area. On the example of Fig. 11, h is equal to two. Thus, the constrained 3D points are located on triangles that are not too close the ones from the others.

The scalar u_{ij} characterizes the elevation of the 3D point $\mathbf{P}_{ij}$ according to the image local reference frame $(\mathbf{O}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, being $\boldsymbol{z}$ the optical axis. $\Delta x_{ij,h}$ and $\Delta y_{ij,h}$ correspond to the 3D steps of the grid, respectively according to the $\boldsymbol{x}$ and $\boldsymbol{y}$ directions. $\Delta x_{ij,h}$ and $\Delta y_{ij,h}$ are constant during the deformation. These quantities are defined by the following equations:

$$\begin{cases} \mathbf{P}^0_{ij}\mathbf{P}_{ij} \cdot \boldsymbol{x} = 0 \\ \mathbf{P}^0_{ij}\mathbf{P}_{ij} \cdot \boldsymbol{y} = 0 \\ \mathbf{OP}_{ij} \cdot \boldsymbol{z} = u_{ij} \end{cases} \quad \text{and} \quad \begin{cases} \Delta x_{ij,h} = \mathbf{P}^0_{i-h,j}\mathbf{P}^0_{i+h,j} \cdot \boldsymbol{x} \\ \Delta y_{ij,h} = \mathbf{P}^0_{i,j-h}\mathbf{P}^0_{i,j-h} \cdot \boldsymbol{y} \end{cases} \tag{8}$$

with $\mathbf{P}^0_{ij}$ the initial position of $\mathbf{P}_{ij}$ on the coarse mesh, i.e. before the deformation. The position of $\mathbf{P}_{ij}$ over its triangle $\mathbf{X}^k_{ij}\mathbf{X}^m_{ij}\mathbf{X}^n_{ij}$ is defined through the use of barycentric coordinates $(\alpha^k_{ij}, \alpha^m_{ij}, \alpha^n_{ij})$ supposed to be constant during the deformation and computed from the initial position $\mathbf{P}^0_{ij}$ of $\mathbf{P}_{ij}$:

$$\begin{cases} \alpha^k_{ij}\mathbf{P}_{ij}\mathbf{X}^k_{ij} + \alpha^m_{ij}\mathbf{P}_{ij}\mathbf{X}^m_{ij} + \alpha^n_{ij}\mathbf{P}_{ij}\mathbf{X}^n_{ij} = \mathbf{0}, \\ \text{with } \alpha^k_{ij} + \alpha^m_{ij} + \alpha^n_{ij} = 1. \end{cases} \tag{9}$$

Finally, for the considered pixel $\mathbf{p}_{ij}$, writing Eqs. (7), (8) and (9) gives rise to a set of non-linear equations whose unknowns are the 3D positions of the nodes of the triangles affected by the projections of the five pixels $\mathbf{p}_{i,j}$, $\mathbf{p}_{i-h,j}$, $\mathbf{p}_{i+h,j}$, $\mathbf{p}_{i,j-h}$ and $\mathbf{p}_{i,j+h}$. Depending on the step value h, the involved triangles can be more or less distant, thus creating a more or less local constraint.

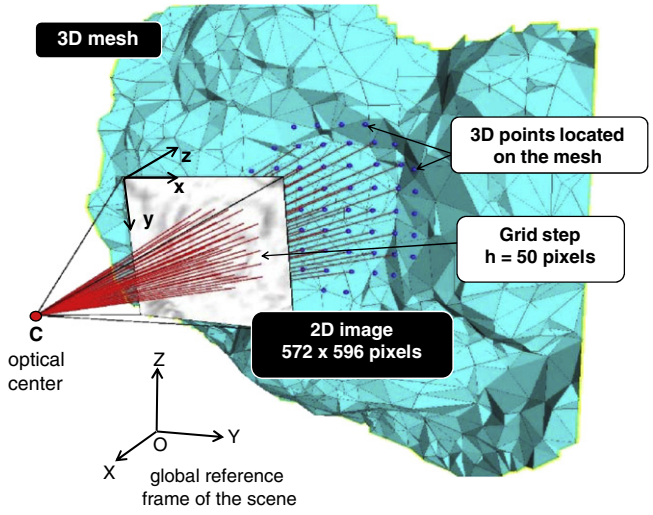


Fig. 12. SFS constraints directly expressed onto the mesh through a set of pixels located on a Cartesian grid of the 2D image (step $h = 50$ pixels between the rows and columns).

4.4. SFS constraints from a single image

Once the hole contour has been reverse-projected onto the 2D image, all the pixels inside the closed polygon can be candidate to the SFS constraint specification (Section 4.3). Since the number of inserted mesh vertices is typically smaller than the number of identified pixels, solely a subset of these inner pixels are used. This is to avoid over-constrained configurations. The various identification algorithms can:

- choose randomly a subset of pixels and use the parameter h to access its neighbours,
- choose the subset according to the value of the brightness function $I(x_i, y_j)$, e.g. when this function is maximum,
- select the subset of pixels aligned on a Cartesian grid whose step is equal to h, the parameter introduced to define the extent of the constraint influence area (see Fig. 12 with $h = 30$).

4.5. SFS constraints from multiple images

The specification of SFS constraints from a calibrated image gives rise to a set of non-linear equations expressed directly according to the 3D position of the inserted coarse mesh vertices. Thus, it is possible to use several calibrated images taken from different view points. The reverse projection of the hole contour in each image gives rise to several sets. These sets can be treated following one of the previously introduced identification algorithm. Each identified pixel produces its own SFS constraints written according to the common unknowns of the problem, i.e. the position of the inserted mesh vertices.

5. Optimization problem formulation and resolution

This section introduces the linear mechanical model of bar networks used to approximate the mesh curvature variation. It takes part to the definition of the optimization problem whose solution satisfies the previously introduced SFS constraints while ensuring curvature blending conditions between the inserted and surrounding meshes.

5.1. Mechanical model of bar networks

First, let us briefly introduce the adopted mechanical model on the example of Fig. 13. Considering an articulated structure, the

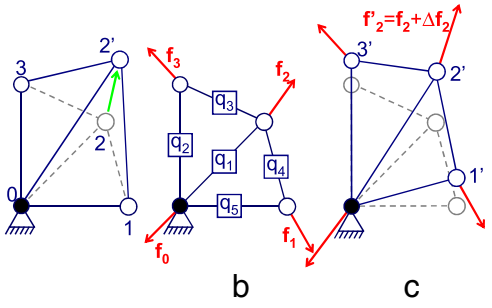


Fig. 13. Deformation of a network with (c) and without (a) a coupling to our mechanical model (b).

positioning of the vertices requires multiple manual displacements which may slow down the overall modification process (Fig. 13(a)). To save time and automate the deformation process, we use a mechanical model of bar networks [52] coupled to the structure. The vertices and edges of the structure 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[1]). To preserve the static equilibrium state of the structure, external forces f_i have to be applied to the nodes (Fig. 13(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 Fig. 13(c), solely the external force applied to the second node is modified to produce the displacement of all the free nodes (see [53] for a complete survey of free-form deformation techniques).

In this paper, we intend to use these external forces to simulate the mesh curvature evolution throughout the objective function to be minimized (Section 5.2). To this aim, the bar network is coupled to the inserted and surrounding meshes. The vertices and edges of the meshes match respectively the nodes and the bars of our bar network.

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 a mesh, 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:

$$\begin{aligned} \boldsymbol{f}_x &= ({}^t\mathbf{C} \cdot \mathbf{Q} \cdot \mathbf{C}) \cdot \boldsymbol{x}, \\ \boldsymbol{f}_y &= ({}^t\mathbf{C} \cdot \mathbf{Q} \cdot \mathbf{C}) \cdot \boldsymbol{y}, \\ \boldsymbol{f}_z &= ({}^t\mathbf{C} \cdot \mathbf{Q} \cdot \mathbf{C}) \cdot \boldsymbol{z}, \end{aligned} \tag{10}$$

where $\mathbf{Q}$ is the force density matrix of size $(N_b \times N_b)$ being N_b the number of bars. $Q_{ij} = q_j \cdot \delta_{ij}$ with δ_{ij} the Kronecker 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.

To specify the degrees of freedom of the bar network, the nodes can either be free or blocked (Fig. 13). Thus, a distinction between these two categories can be performed in Eq. (10). It gives rise to the two following sets of equations:

$$\begin{aligned} \boldsymbol{f}_{fn_x} &= \mathbf{D}_f \cdot \boldsymbol{x}_{fn} + \mathbf{D}_{bf} \cdot \boldsymbol{x}_{bn}, \\ \boldsymbol{f}_{fn_y} &= \mathbf{D}_f \cdot \boldsymbol{y}_{fn} + \mathbf{D}_{bf} \cdot \boldsymbol{y}_{bn}, \\ \boldsymbol{f}_{fn_z} &= \mathbf{D}_f \cdot \boldsymbol{z}_{fn} + \mathbf{D}_{bf} \cdot \boldsymbol{z}_{bn}, \end{aligned} \tag{11}$$

[1] This approach is also called the Force Density Method (FDM).

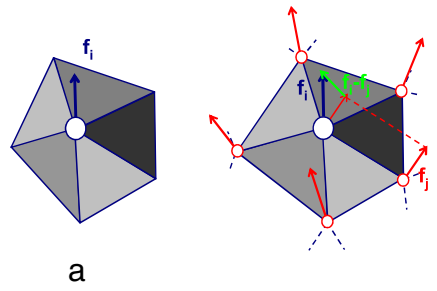


Fig. 14. External forces taken into account when minimizing either the force applied to the ith node (a) or the variations of this force with respect to the surrounding ones (b).

for the external forces applied to free nodes and:

$$\begin{aligned} \boldsymbol{f}_{bn_x} &= {}^t\mathbf{D}_{bf} \cdot \boldsymbol{x}_{fn} + \mathbf{D}_b \cdot \boldsymbol{x}_{bn}, \\ \boldsymbol{f}_{bn_y} &= {}^t\mathbf{D}_{bf} \cdot \boldsymbol{y}_{fn} + \mathbf{D}_b \cdot \boldsymbol{y}_{bn}, \\ \boldsymbol{f}_{bn_z} &= {}^t\mathbf{D}_{bf} \cdot \boldsymbol{z}_{fn} + \mathbf{D}_b \cdot \boldsymbol{z}_{bn}, \end{aligned} \tag{12}$$

for the external forces applied at blocked nodes. The different $\mathbf{D}_i$ matrices are obtained through the decompositions of the ${}^t\mathbf{C} \cdot \mathbf{Q} \cdot \mathbf{C}$ matrix as detailed in [53,54].

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:

$$\begin{aligned} \boldsymbol{x}_{fn} &= (\mathbf{D}_f)^{-1} \cdot (\boldsymbol{f}_{fn_x} - \mathbf{D}_{bf} \cdot \boldsymbol{x}_{bn}), \\ \boldsymbol{y}_{fn} &= (\mathbf{D}_f)^{-1} \cdot (\boldsymbol{f}_{fn_y} - \mathbf{D}_{bf} \cdot \boldsymbol{y}_{bn}), \\ \boldsymbol{z}_{fn} &= (\mathbf{D}_f)^{-1} \cdot (\boldsymbol{f}_{fn_z} - \mathbf{D}_{bf} \cdot \boldsymbol{z}_{bn}). \end{aligned} \tag{13}$$

These last equations show how it is possible to manipulate indirectly the node positions, and consequently the inserted mesh vertices, through the manipulation of external forces (see [53] for the treatment of configurations where the $\mathbf{D}_f$ matrix is singular). The unknowns of the deformation process are not longer the positions but the external forces themselves. Even if only the external forces applied at the free nodes are necessary to compute the new free nodes positions, some external forces applied to the 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 5.2). They are linearly dependent on the external forces applied at free nodes and the corresponding equations can be obtained using a combination of Eqs. (12) and (13).

5.2. Curvature variation minimization

When considering solely the blending conditions, i.e. without the SFS constraints, the optimization problem is reduced to the minimization of an objective function. Theoretically, any combination of quantities, expressed as a function of the free nodes position, can be used to define this single objective function [54]. To speed up the resolution process, we have been considering quadratic functions. Thus, their minimization gives rise to a system of linear equations that can be solve easily (no need of an iterative process). Here, the function to be minimized uses our mechanical model of bar network linking the nodes position to the so-called external forces (Eqs. (12) and (13)).

Among the possible quadratic functions, the simplest one corresponds to the sum of all the external forces applied on the mesh vertices:

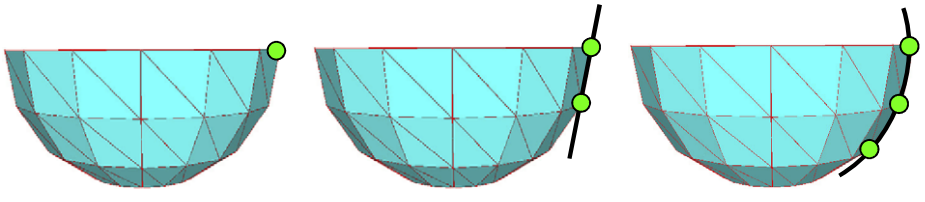
$$\phi_F = \sum_i \boldsymbol{f}_i^2 = \sum_i f_{x_i}^2 + f_{y_i}^2 + f_{z_i}^2. \tag{14}$$

Since the external forces applied to the free nodes are independent of each other (Fig. 14(a)), the minimization of this sum on all the free nodes always produces a null unknown vector (no crossed terms in the quadratic function). This tends to minimize the curvature of the underlying geometry while producing areas as planar as possible (Fig. 16(a)). This minimization is interesting in the context of tensile textile structures. Moreover, one can notice that an external force is only influenced by both the position of the node on which it is applied and the position of the vertices directly connected to it (Fig. 14(a)). Therefore, solely the information relative to the position of the hole contour vertices are considered. 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 in the sum of Eq. (14) the external forces applied to the nodes of the hole contour. Using a combination of Eqs. (12) and (13), 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 hole contour and to the second row of the surrounding mesh, i.e. relative to all the vertices that are connected to the vertices of the hole contour (Fig. 16(b)). Unfortunately, such a minimization cannot succeed in more than preserving the tangency with the surrounding mesh (Fig. 15(b)).

To reach the level of curvature blending condition, one should use geometric information relative to the three first rows of the surrounding mesh (Fig. 15(c)). While noticing that the variation between two successive external forces represents in some sense the evolution of the curvature between the two nodes, the following minimization can be adopted:

$$\phi_{\Delta^r F} = \sum_i \sum_j \left[\boldsymbol{f}_i - \boldsymbol{f}_j\right]^2 \tag{15}$$

where the sum on i is performed on all the nodes and the sum on j is restricted to the nodes connected to node i. The use of such a functional enables the minimization of the variations between couples of external forces applied to nodes directly connected together. The influence area of these basic quantities is greater than in the previous case (Fig. 16(c)). This is due 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. 14(b)) and that each external force is influenced by both the position of the node on which it is applied and the position of the vertices directly connected to it. But our initial requirements are still not satisfied if the minimization of this sum is performed on all the free nodes (Fig. 16(c)). In this



(a) Position. (b) Tangency. (c) Curvature.

Fig. 15. Depending on the number of vertex rows taken into account, some position or tangency or curvature information can be extracted from the surrounding mesh.

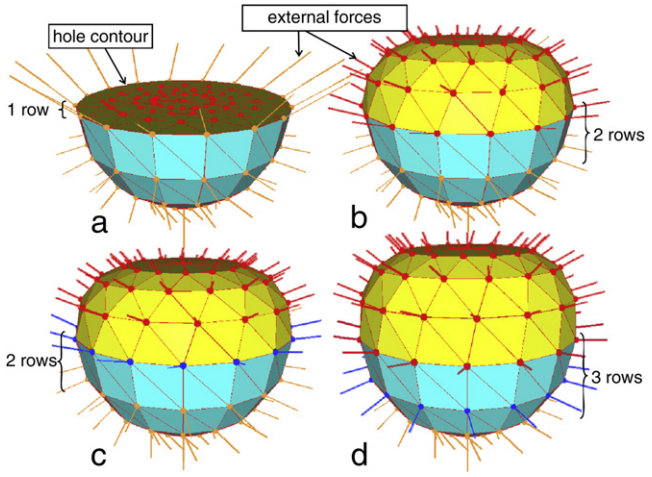


Fig. 16. Minimization of ϕ_F applied to free nodes and to nodes of the hole contour (a, b). Minimization of $\phi_{\Delta^r F}$ applied to free nodes and to nodes of the hole contour (c,d).

case, solely the two first rows of the surrounding mesh are used. To take into account the third row, the sum on i also has to take into account the nodes of the hole contour (Fig. 16(d)). In this case, the evolution of the curvature is preserved when crossing the hole contour. More exactly, the curvature variation minimization is approximated by the external forces relative variation minimization.

Finally, one can notice that there is no need to build the mechanical model on the whole surrounding mesh since solely the external forces applied to the two first rows of vertices are used. Thus, in the proposed approach, the mechanical model is built both on the inserted mesh and on the three first rows of the surrounding mesh. All the nodes coupled to the surrounding mesh are blocked to preserve its shape.

5.3. Optimization problem resolution

In this section, we assume that the hole has been filled in by a coarse mesh and that a set of N_p pixels have been identified in a calibrated image. A bar network is coupled to the inserted mesh as well as to the three first rows of the surrounding mesh. All the nodes surrounding the inserted mesh are blocked. It remains N_{fn} free nodes whose position is unknown and whose coordinates are stored within the three vectors $\boldsymbol{x}_{fn}, \boldsymbol{y}_{fn}$ and $\boldsymbol{z}_{fn}$. Following Section 4, we define $3 \times N_p$ equations inside the global constraints vector $\boldsymbol{G}$. They are either linearly or non-linearly dependent of the $3 \times N_{fn}$ unknowns:

$$\boldsymbol{G}(\boldsymbol{x}_{fn}, \boldsymbol{y}_{fn}, \boldsymbol{z}_{fn}) = \boldsymbol{0}. \tag{16}$$

Using the quadratic function defined in Eq. (15), we set up the minimization of the external forces relative variation including the nodes of the hole contour:

$$\min \phi(\boldsymbol{f}_{fn_x}, \boldsymbol{f}_{fn_y}, \boldsymbol{f}_{fn_z}, \boldsymbol{f}_{bn_x}, \boldsymbol{f}_{bn_y}, \boldsymbol{f}_{bn_z}). \tag{17}$$

Finally, we use Eqs. (11) and (12) in Eq. (17) to formulate the optimization problem with respect to the free nodes position unknown vectors:

$$\begin{cases} \boldsymbol{G}(\boldsymbol{x}_{fn}, \boldsymbol{y}_{fn}, \boldsymbol{z}_{fn}) = \boldsymbol{0} \\ \min \phi(\boldsymbol{x}_{fn}, \boldsymbol{y}_{fn}, \boldsymbol{z}_{fn}). \end{cases} \tag{18}$$

One can notice that the proposed optimization problem formulation clearly decouples the minimization relative to the external forces from the constraints issued from the SFS equations. Thus, the specification of the local shape constraints is decoupled from the specification of the overall deformation behaviour. In our implementation, we use the Differential Evolution method [55] to solve this optimization problem made of $3 \times N_p$ equations and $3 \times N_{fn}$ unknowns. Differential evolution is a stochastic parallel direct search evolution strategy optimization method that is fairly fast and reasonably robust.

To avoid over-constrained configurations, the number of considered pixels is smaller than the number of inserted mesh vertices. Thus, there are no limitations to the use of several calibrated images which bring several sets of constraints gathered together in the unique constraint vector $\boldsymbol{G}$.

Table 2
Quantitative comparison of the results obtained by minimizing the curvature variation either solely, or together with the satisfaction of SFS constraints on the 2D curve example.

2D curve model	Curvature variation vs. initial model	Curvature variation + SFS constraints vs. initial model
RMS deviation	0.389	0.012

6. Results and discussion

6.1. Filling a hole in a 2D curve

To illustrate the capabilities of our image-based hole-filling algorithm, we first consider a two-dimensional example of a continuously derivable curve whose brightness function can be computed directly (Fig. 17):

$$u(x) = \frac{1}{\sqrt{1 + (1.3x - 0.35x^3)^2}} \quad \text{with } I(x) = \sqrt{\frac{1}{1 + u'(x)^2}}. \tag{19}$$

An artificial hole has been done in a complex shape area by deleting the curve within the interval $[-1.5, 1.5]$ (Fig. 18(a)). This hole is then filled in by using a straight polyline defined by 11 inner vertices (Fig. 18(b)). To recover the missing area, three tests have been performed:

(1) thirteen SFS constraints are assigned to the eleven inner vertices of the polyline plus the two end points (Fig. 18(c_1)–(c_3)). Since there is an infinite number of solutions, we set the location of certain points and choose arbitrary solutions among the possible locations. Here, solely three solutions are depicted. These three solutions satisfy the same set of SFS constraints. There is no guarantee to obtain a solution that reconstructs well the initial curve and the resolution process is tedious.
(2) the relative variations of the external forces applied to the bar network coupled to the polyline (Section 5.2) are minimized. This tends to produce a shape whose curvature variations are minimal. It produces a smooth connection between the surrounding curve and the inserted polyline, but it fails to reconstruct the form of the complex shape (Fig. 18(d)).
(3) both the thirteen SFS constraints and the minimization of the external forces relative variations are used to find a solution (Fig. 18(e)) whose shape is close to the one initially removed (Fig. 17(a)). Thus, using the proposed approach that combines SFS constraints with a minimization that is based on the curvature variations, the drawbacks of each method are overcome.

To compare the quality and accuracy of the recovery, we use the Root Mean Square Deviation (RMSD) metric. First, the filling obtained by minimizing the curvature variation (Fig. 18(d)) is compared to the initial curve (Fig. 17(a)). The result obtained using the proposed hole-filling algorithm (Fig. 18(d)) is then compared to the initial curve (Fig. 17(a)). As depicted on Table 2, the result obtained by taking into account SFS constraints is far closer to the ground truth than the one obtained by minimizing solely the curvature variation.

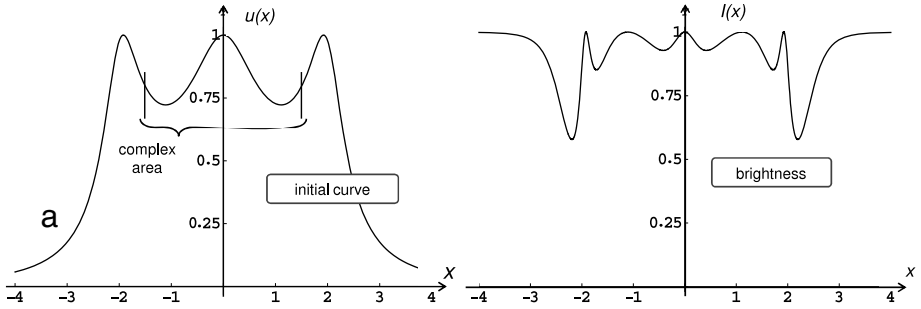


Fig. 17. An analytical 2D initial curve (a) with its brightness evolution (b).

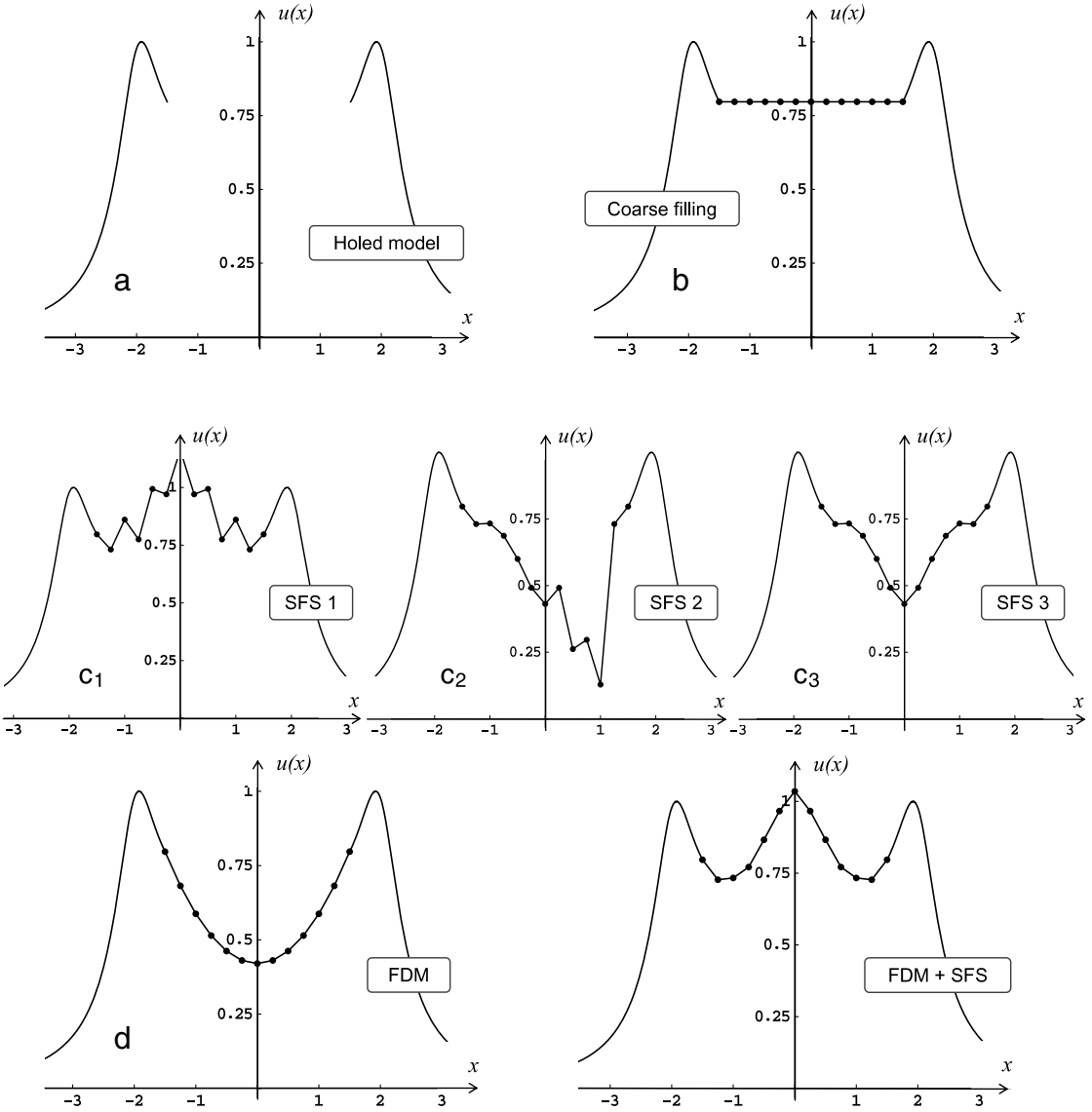


Fig. 18. The initial holed curve (a) filled in with a straight polyline (b) deformed to satisfy either the SFS constraints (c_1), or the minimization of the external forces relative variations (d), or both the SFS constraints and the minimization of the external forces relative variations (e).

6.2. Filling holes in 3D triangle meshes

The second example comes from a polyhedral model of an architectural edifice (Fig. 19(a)). 46 simple holes are detected and treated by using the triangulation algorithm of [13] and the minimization of the curvature variation detailed in Section 5.2 (Fig. 19(b)).

To further test the proposed image-based hole-filling algorithm, a hole has been created (Fig. 20(b)) inside a more complex area (Figs. 19(b) and 20(a)). Once the hole contour identified and

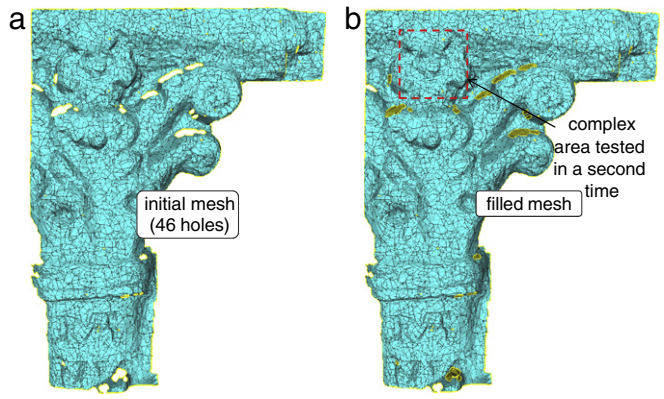


Fig. 19. Polyhedral model of an architectural edifice before (a) and after (b) the filling of 46 holes using solely the minimization of the external forces variations.

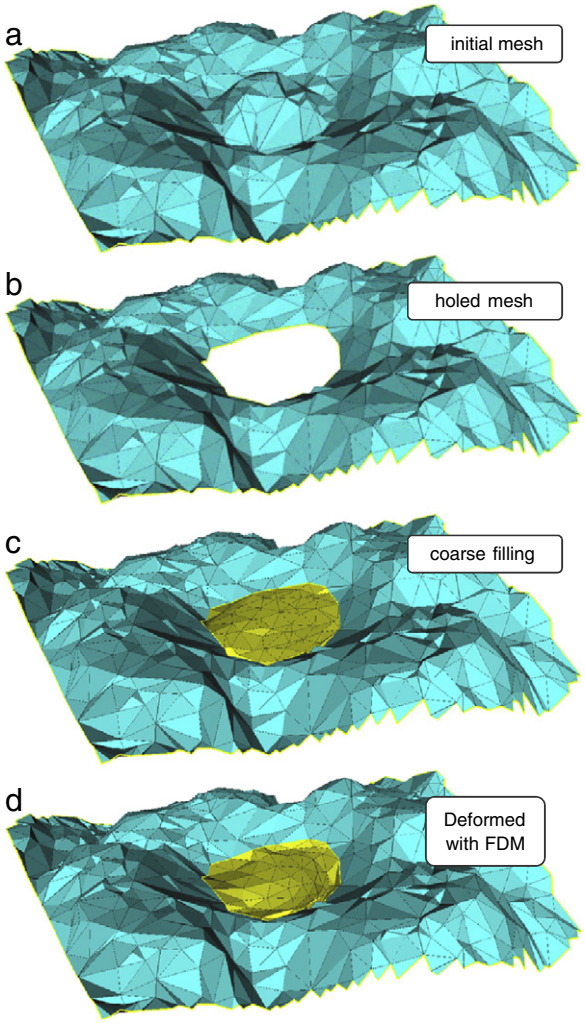


Fig. 20. Filling a hole in a complex area using the Force Density Method (FDM) which tends to minimize the mesh curvature evolution.

cleaned, the hole is first filled in (Fig. 20(c)) following the equilaterality criterion of [13]. The sizes of the inserted triangles match the sizes of the surrounding triangles, and the inserted mesh does not have privileged directions. Therefore, the shape quality criteria introduced in Section 1 are satisfied.

Our bar network is then coupled to the vertices and edges of the mesh (Section 5.1). All the nodes are blocked excepting the 44 nodes newly inserted. These are the free nodes considered as

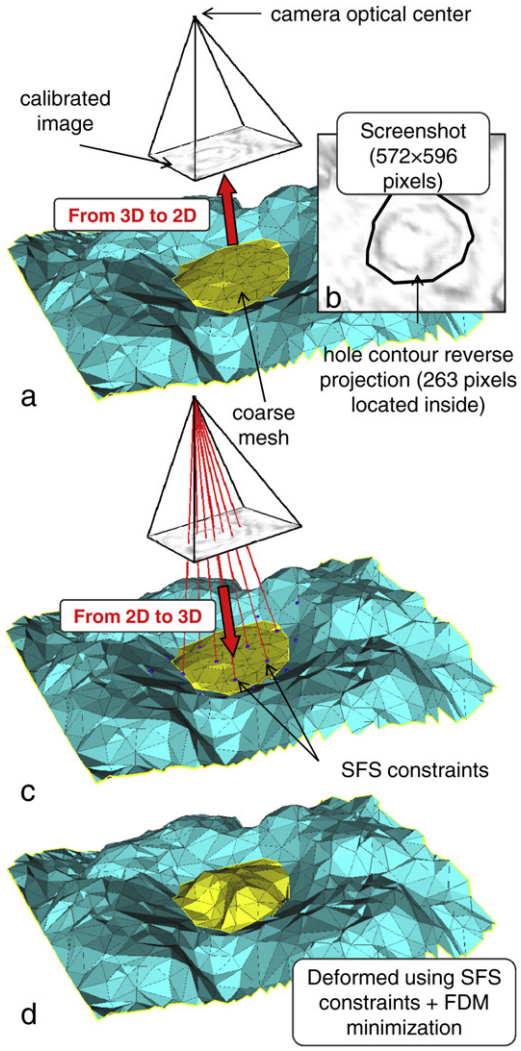


Fig. 21. Filling a hole using the FDM together with SFS constraints.

the unknowns of our deformation process. Fig. 20(d) shows the result of the minimization of the external forces relative variations. The inserted mesh blends smoothly with the surrounding one, which satisfied the second criterion introduced in Section 1, but the complex shape is not recovered.

The image used to recover the complex shape is a screenshot of the initial mesh (Fig. 21(b)). Here, the model is lighted with only one source located at the infinite and whose direction is orthogonal to the screen. Thus, the calibration of the image is automatic (Fig. 21(a)). A reverse projection step enables the identification of the 263 image pixels taking part to the hole contour definition [51]. A grid of 150 pixels pitch is used to define the SFS constraints onto the inserted patch (Fig. 21(c)). It leads to the specification of 48 constraints (10 SFS non-linear constraints and 38 linear constraints) constraining the position of the 44 free nodes. The resolution of this optimization problem, i.e. SFS constraints and minimization of the external forces relative variations, produces a resulting mesh (Fig. 21(d)) that satisfies all the criteria enounced in Section 1.

To compare the quality and accuracy of the recovery, we use the Hausdorff distance metric using the METRO software [56]. First, the Hausdorff distance between the initial triangle mesh (Fig. 20(a)) and the mesh obtained by minimizing solely the curvature variation (Fig. 20(d)). The result obtained with our newly developed image-based hole-filling algorithm (Fig. 20(e)) is then compared to the initial model (Fig. 20(a)). As depicted on Table 3,

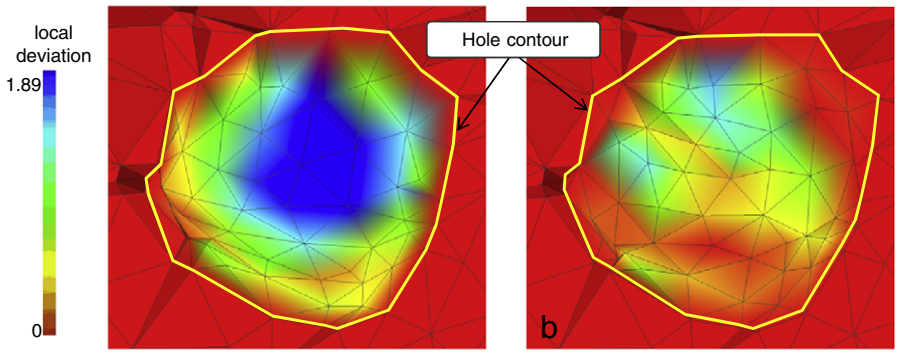


Fig. 22. Comparison of the local deviations between the initial mesh and the result of the minimization of the curvature variation used either solely (a), or together with the specification of SFS constraints (b).

Table 3
Quantitative comparison of the results obtained by minimizing the curvature variation either solely, or together with the satisfaction of SFS constraints on the architectural edifice model.

Architectural edifice model	Curvature variation vs. initial model	Curvature variation + SFS constraints vs. initial model
Hausdorff distance	3.045	1.883

the result obtained with our image-based hole-filling algorithm is far closer to the ground truth than the one obtained by minimizing solely the curvature variation. Fig. 22 shows the distribution of the local deviations between the meshes in both cases.

The algorithm has been tested on an Intel Core 2 duo, 3 GHz CPU, 8 GB RAM. In the actual implementation, we use Mathematica 6 to solve the optimization problem with the Differential Evolution method [55]. Since the commands are interpreted at the time of the evaluation, the resolution process is slowed down thus affecting significantly the computation times. On the example of the architectural edifice, for a relative error on the position of the vertices of 10^{-2}, the resolution method converges in about 20 iterations thus taking 49 s. In the future, we planned to use optimization packages directly implemented in C/C++ as the other parts of our prototype software.

Integrating in a common framework the SFS technology together with our mesh deformation technique seems promising. We obtain good results with respect to the various criteria introduced in Section 1. However, several improvements have been envisaged. Among them, there are:

- the use of perspective SFS constraints: indeed, the specified SFS constraints constrain the projected points to move along a direction $\boldsymbol{z}$ orthogonal to the calibrated image. Actually, the use of perspective constraints could be more interesting, especially if large deformations are required.
- the relaxation of constraints: in the implemented version, the barycentric coordinates of the points that have been projected on the mesh stay constant during the deformation process. This may lead to undesired behaviours, especially if large deformations are required, or if the deformation cannot be obtained in one step. Since the resolution process is iterative (non-linear constraints), we plan to relax the barycentric coordinates by reprojecting the pixels onto the mesh obtained at each step.
- the capacity to take into account constraints arising from multiple digital images: this should help the recovery of much more complex shapes such as mushrooms. Actually, this should not be difficult to obtain since the proposed framework relies on an optimization problem whose unknowns are the free nodes position (the same for each image) and whose constraints directly link these unknowns to the values of the brightness functions (different for each image).

7. Conclusion and future works

In this paper, a set of models, methods and tools has been proposed to fill in holes in meshes. The solution is obtained by solving an optimization problem formed by both a set of non-linear equations coming from the Shape From Shading technologies, and a functional to be minimized in order to preserve the curvature evolution between the inserted and surrounding meshes. The function to be minimized is quadratic with respect to the parameters of the linear mechanical model of bar network that is used. Hence, the limitations of the SFS technologies and deformation techniques are overcome. The resulting mesh satisfies curvature blending conditions thus circumventing concave/convex ambiguities. Its inserted shape also mimics the potentially complex shape evolutions that are extracted from 2D digital images. Finally, the resulting mesh does not own privileged directions and can have a greater number of vertices than the number of pixels contained into the considered area. This is really different from what can be found in the literature.

However, some new achievements can still be considered. More precisely, some issues or parameters which can affect the shape of the resulting patch, are not fully automatically computed yet. They are related to the defined SFS constraints and to:

(1) the satisfaction of the ideal SFS assumptions, which means the position and the orientation of the camera as well as the material properties of the physical object must be known. Moreover, they have to check specific criteria as explained in Section 4.
(2) the number of imposed SFS constraints which can be determined automatically or left to the user through the parameter h (see Section 4.4). The more h is low, the more the patching mesh will be constrained.
(3) the kind of imposed SFS constraints which are obtained using orthogonal projection. Relaxing them during the iterative deformation process by the use of perspective projection computed at each iteration could help and speed up the resolution of the optimization problem (see Section 6.2).

Finally, one of the big challenges remains the use of the proposed approach on real noisy images whose lighting source is unknown.

Other interesting applications such as the copy/paste of a shape from mesh to mesh, or from image to mesh, are envisaged. In an industrial context, we are also working on the adaptation of our technique to the update of a digital model of a structure/edifice according to information retrieved from 2D images. This is interesting when the digital model is obsolete, i.e. when the real structure/edifice has partially evolved since the acquisition.

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, 2004. URL: http://www.aimatshape.net.

[2] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on computer graphics. 1992. p. 71–8.

[3] Edelsbrunner H, Mücke EP. Three-dimensional alpha shapes. ACM Transactions on Graphics 1994;13(1):43–72.

[4] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques, vol. 32. 1998. p. 415–21.

[5] Bernardini F, Mittleman J, Rushmeier H, Silva C, Taubin G. The ball-pivoting algorithm for surface reconstruction. IEEE Transactions on Visualization and Computer Graphics 1999;5(4):349–59.

[6] Ma W, Kruth JP. Parameterization of randomly measured points for least squares fitting of *B*-spline curves and surfaces. Computer-Aided Design 1995; 27(9):663–75.

[7] Forsey DR, Bartels RH. Surface fitting with hierarchical splines. ACM Transactions on Graphics 1995;14(2):134–61.

[8] Eck M, Hoppe H. Automatic reconstruction of *B*-spline surfaces of arbitrary topological type. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques, vol. 30. 1996. p. 325–34.

[9] Ma W, Zhao N. Catmull–Clark surface fitting for reverse engineering applications. In: Proceedings of geometric modeling and processing, theory and applications. 2000. p. 274–83.

[10] Lee A, Moreton H, Hoppe H. Displaced subdivision surfaces. In: Siggraph 2000, Computer graphics proceedings. 2000. p. 85–94.

[11] Takeuchi S, Kanai T, Suzuki H, Shimada K, Kimura F. 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. 2000. p. 202–12.

[12] Ciarlet PG. The finite element method for elliptic problem. North Holland; 1978.

[13] Lou R, Mikchevitech A, Pernot J-P, Véron P. Direct merging of triangle meshes preserving simulation semantics for fast modification of numerical models. In: Tools and methods for competitive engineering, vol. 1. 2008. p. 119–31.

[14] De Luca L. Relevé et multi-représentations du patrimoine architectural: Définition dune approche hybride de reconstruction 3D dédifices. Ph.D. thesis. ENSAM. 2006. Directeurs: P. Véron et M. Florenzano.

[15] Curless B, Levoy M. A volumetric method for building complex models from range images. In: Processing of SIGGRAPH'96. 1996. p. 303–12.

[16] Davis J, Marschner SR, Garr M, Levoy M. Filling holes in complex surfaces using volumetric diffusion. In: Processing of the first international symposium on 3D data, visualization and transmission. 2002. p. 428–38.

[17] Nooruddin FS, Turk G. Simplification and repair of polygonal models using volumetric techniques. IEEE Transactions on Visualization and Computer Graphics 2003;9(2):191–205.

[18] Ju T. Robust repair of polygonal models. In: SIGGRAPH '04: ACM SIGGRAPH 2004 papers. New York (NY, USA): ACM; 2004. p. 888–95.

[19] Verdera J, Caselles V, Bertalmio M, Sapiro G. Inpainting surface holes. In: International conference on image processing. 2003. p. 903–6.

[20] Clarenz U, Diewald U, Dziuk G, Rumpf M, Rusu R. A finite element method for surface restoration with smooth boundary conditions. Computer Aided Geometric Design 2004;21(5):427–45.

[21] Xu G, Pan Q, Bajaj CL. Discrete surface modelling using partial differential equations. Computer Aided Geometric Design 2006;23(2):125–45.

[22] Liepa P. Filling holes in meshes. In: Proceedings of the 2003 eurographics/ACM SIGGRAPH symposium on geometry processing. 2003. p. 200–5.

[23] Barequet G, Sharir M. Filling gaps in the boundary of a polyhedron. Computer Aided Geometric Design 1995;12(2):207–29.

[24] Pfeifle R, Seidel H-P. Triangular *B*-splines for blending and filling of polygonal holes. In: Proceedings of graphics interface. 1996. p. 186–93.

[25] Kobbelt L, Campagna S, Vorsatz J, Seidel H-P. Interactive multi-resolution modeling on arbitrary meshes. In: Proceedings of SIGGRAPH'98. 1998. p. 105–14.

[26] Schneider R, Kobbelt L. Geometric fairing of irregular meshes for free-form surface design. Computer-Aided Geometric Design 2001;18(4):359–79.

[27] Desbrun M, Meyer M, Schroder P, Barr AH. 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. 1999. p. 317–24.

[28] Wang J, Oliveira MM. A hole-filling strategy for reconstruction of smooth surfaces in range images. In: Proceedings of SIBGRAPI'03. 2003. p. 11–8.

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

[30] Pernot J-P, Moraru G, Véron P. Filling holes in meshes using a mechanical model to simulate the curvature variation minimization. Computers & Graphics 2006;30(6):892–902. ISSN 0097-8493.

[31] Pernot J-P, Moraru G, Véron P. Filling holes in meshes for efficient reverse engineering of products. In: Tools and methods for competitive engineering. 2006. p. 273–84.

[32] Teng C-H, Chen Y-S, Hsu W-H. Constructing a 3D trunk model from two images. Graphical Models 2007;69(1):33–56.

[33] Criminisi A, Zisserman A. Shape from texture: Homogeneity revisited. In: Proc. 11th British machine vision conference. 2000. p. 82–91.

[34] Yu Y, Chang JT. Shadow graphs and surface reconstruction. In: In European conference on computer vision. Springer-Verlag; 2002. p. 31–45.

[35] Fromherz T, Bichsel M. Shape from contours as initial step in shape from multiple cues. In: Proceedings of the ISPRS commission III symposium on spatial information from digital photogrammetry and computer vision. 1994. p. 249–56.

[36] Horn BKP, Winston PH. Obtaining shape from shading information. In: The psychology of computer vision. McGraw-Hill; 1975. p. 115–55.

[37] Belhumeur PN, Kriegman DJ, Yuille AL. The bas-relief ambiguity. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 1997. p. 1060–6.

[38] Rouy E, Tourin A. A viscosity solutions approach to shape-from-shading. SIAM Journal on Numerical Analysis 1992;29(3):867–84.

[39] Prados E, Faugeras O, Rouy E. Shape from shading and viscosity solutions. In: Proceedings of the 7th European conference on computer vision (volume II). Copenhagen, Denmark. Lecture notes in computer science, vol. 2351. 2002. p. 790–804.

[40] Daniel P, Durou J-D. From deterministic to stochastic methods for shape from shading. In: Proceedings of the 4th Asian conference on computer vision. 2000. p. 187–92.

[41] Durou J-D, Falcone M, Sagona M. A survey of numerical methods for shape from shading. Technical report. IRIT. 2004.

[42] Prados E, Faugeras O. Shape from shading. In: Paragios N, Chen Y, Faugeras O, editors. Handbook of mathematical models in computer vision. Springer-Verlag; 2005. p. 377–90.

[43] Turk G, Levoy M. Zippered polygon meshes from range images. In: SIGGRAPH'94: Proceedings of the 21st annual conference on computer graphics and interactive techniques. New York (NY, USA): ACM; 1994. p. 311–8.

[44] Fua P, Leclerc YG. Taking advantage of image-based and geometry-based constraints to recover 3-D surfaces. Computer Vision and Image Understanding 1995;64(1):111–27.

[45] Pernot J-P, Moraru G, Véron P. Repairing triangle meshes built from scanned point cloud. Journal of Engineering Design 2007;18(5):459–73. ISSN 0954-4828.

[46] Horn B, Brooks MJ. Shape from shading. MIT Press; 1989.

[47] Bakshi S, Yang Y-H. Shape from shading for non-lambertian surfaces. In: ICIP (2). 1994. p. 130–4.

[48] Horaud R, Monga O. Géométrie et calibration. Hermès edition; 1995.

[49] Panchetti M, Pernot J-P, Véron P. Polyhedral simplifications preserving character lines extracted from images. In: Proceedings of the IEEE shape modelling international conference. 2007. p. 81–90.

[50] Faugeras OD, Luong Qt. Camera self-calibration: Theory and experiments. Springer-Verlag; 1992. p. 321–34.

[51] Haines E. Point in polygon strategies. In: Heckbert Paul, editor. Graphics gems IV. Academic Press; 1994. p. 24–46.

[52] Schek HJ. The force density method for form finding and computation of general networks. Computer Methods in Applied Mechanics and Engineering 1974;3(2):115–34.

[53] Pernot J-P. Fully free form deformation features for aesthetic and engineering designs. Ph.D. thesis. France: INP-Grenoble. Italy: University of Genoa; 2004.

[54] Pernot J-P, Guillet S, Léon J-C, Falcidieno B, Giannini F. Shape tuning in fully free form deformation features. Journal of Computing and Information Science in Engineering (JCISE) 2005;5(1):95–103.

[55] Storn R, Price K. Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. Journal of Global Optimization 1997;11:341–59.

[56] Cignoni P, Rocchini C, Scopigno R. Metro: Measuring error on simplified surfaces. Computer Graphics Forum 1998;17(2):167–74.