

Surface Reconstruction and Geometric Modeling for Digital Prosthesis Design

Luiz C. M. de Aquino and Gilson A. Giraldi
(Aff) National Laboratory for Scientific Computing.
Petrópolis, Brazil.

Email: {lcaquino, gilson}@lncc.br

Paulo S. S. Rodrigues

(Aff) FEI University, Department of Computer Science.
São Bernardo do Campo, Brazil.

Email: psergio@fei.edu.br

Antônio Lopes A. Junior

(Aff) Feira de Santana State University.

Feira de Santana, Brazil.

Email: alopesajr@gmail.com

Jaime S. Cardoso

(Aff) INESC Porto, Faculty of Engineering, University of Porto.
Porto, Portugal.

Email: jaime.cardoso@ieee.org

Jasjit Suri

(Aff) Biomedical Technologies, Inc.

Denver, CO, USA

&

(Aff) Idaho State University

ID, USA

Email: jsuri@comcast.net

Abstract. The restoration and recovery of a defective skull can be performed through operative techniques to implant a customized prosthesis. Recently, image processing, surface reconstruction and geometric methods have been used for digital prosthesis design. In this chapter we review state-of-the-art approaches in this field and discuss related issues. The field of prosthesis modeling may include methods for segmentation and surface reconstruction, geometric modeling, multiscale methods and user interaction approaches. So, we present the background in the area by reviewing methods in isosurface extraction from 3D images, deformable models, wavelets and subdivision surfaces. Then, we discuss some proposals in this area: a balloon model for slice-by-slice bone reconstruction, the T-Surfaces plus isosurface generation models as a general framework for surface reconstruction and user interaction, wavelets-based multiscale methods and filling holes methods. We describe also experimental results and a computational tool that we are developing for image processing and visualization which can be used for digital prosthesis design. We offer a discussion by presenting some perspectives and issues related to

the models described on previous sections. Finally, we present the conclusions of our work.

1 Introduction

Mending an imperfect skull is necessary not just for cosmetic reasons but also since sizable imperfections might endanger a fundamental section of the brain. Cranioplasty (the procedure of mending imperfections in the cranium with cranial engrafts) is feasible in these cases [38]. However, cranioplasty continues to be a challenge to craniofacial surgeons and neurosurgeons which motivates the development of computational tools for surgical planning [10]. It is even more difficult when there is great damage or if it occurs in an area where the structure is fragile or if it involves critical tissues like the eyes.

So far, a common procedure has not been developed for the conceptualization and fabrication of pre-manufactured cranial engrafts. Many techniques exist but the treatment is differentiated according to the methodology specifics. Furthermore, each technique fulfills personal customization differently [29, 46].

The use of medical image processing, computer graphics and rapid prototyping (RP) technology permits personal customization to decrease patients' injuries during surgeries [23]. The creation of the prosthesis starts with a computed tomography (CT) of the cranium. Next, image processing methods are implemented for segmentation. In addition, a surface reconstruction technique is implemented in order to acquire a three-dimensional model of the prosthesis, which will become the insert for the following stage when the prosthesis is fabricated using RP technology. Finally, the cranial imperfection is replaced by the prosthesis [29], [46], [48].

There are two categories of approaches in this field, in relation to the image processing and geometric modeling. In the first category, surface rebuild is applied in order to have a representation of the imperfection, which is used later to create the digital model of the missing piece. In [49] we find such method which divides each voxel into 5 tetrahedra and apply the marching cubes technique to get a polygonal representation of the defective skull. Then, the method takes user-defined guiding points and apply computational geometry algorithms to generate the digital version of the prosthesis. In this process, the geometric model of the prosthesis is obtained taking into account only the polygonal mesh.

The second category includes techniques which rebuild the cranium digitally. Moreover, Boolean operations and the marching cubes technique are employed to acquire the prosthesis' digital model. The technique of reflection, built on the assumption of cranium symmetry, is part of this category. From this point, the axis of symmetry is calculated in order to reflect the segmented image in respect to this axis. However, this technique may only be used if the injury does not intersect that axis. If it does, then we must rebuild the missing piece according to the geometry of the cranium surrounding the injury and the soft tissues of the brain. The separation of bone is achieved without difficulty through thresholding techniques. But, separation of soft tissues is not easily accomplished.

For example in [33] there is a case study of a young woman who sustained significant skull damage from a car accident. In her case, computational techniques for quickly creating a customized prosthesis aided the doctors, who used Mimics software [3] to search for a threshold to separate the bone using the CT scan information.

The doctors needed to remove the meninges to make sure that the inside of the engraft did not obstruct the meninges and also to acquire another reference for the creation of the prosthesis. Next, another software, the 3-Matic [1] was used to draw guiding curves on the usable references (part of the meninges and bone particles) to indicate the necessary curvature of the implant. These curved lines complied to a convex path over the notch of the meninges to preclude the prosthesis from applying force to the meninges.

Because MRI is better at differentiating soft tissue; it is generally preferred over CT for brain imaging which makes the research on CT brain segmentation relatively scarce. However, for prosthesis design, MRI is contraindicated because the main target is the bone tissue. In [24] it is presented a study that considers the effectiveness of existing algorithms for segmenting brain tissue in CT images. Three methods (Bayesian classification, Fuzzy c-Means and Expectation Maximization) were used to segment brain and cerebrospinal fluid. The results shown that these methods outperformed commonly used threshold-based segmentation and points towards the necessity for new imaging protocols for optimizing CT imaging for soft tissue scanning.

Aside from segmentation and image processing, problems with geometric models may occur. Surface reconstruction, filling holes techniques, and multi-scale methods must be considered in this venue.

Among the isosurface methods [43], two types are considered in this paper: continuation and marching methods. Continuation methods propagate the surface from a set of seed cells. The key idea is to use the spatial coherence of the isosurface during its extraction [43, 8]. In Marching Cubes, each surface-finding phase visits all cells of the volume, normally by varying coordinate values in a triple “for” loop [30]. As each cell that intersects the isosurface is found, the necessary polygon(s) to represent the portion of the isosurface within the cell is generated.

The reconstruction of the lost part of the bone can be cast in the problem of constructing a *n-sided* surface patch that smoothly connects the surfaces that surrounds the polygonal hole. According to [27], *n-sided* patches generation algorithms falls basically into two classes. In the first class, the polygonal domain is subdivided in the parametric plane. So, triangular or rectangular elements are put together or recursive subdivision methods are applied. In the second class, one uniform equation is used as a combination of 3D constituents. In this case the surface equation can be computed by either generalized control-point based methods or through a weighted sum of 3D interpolants.

On the other hand, multiscale methods may be applied: having detected a structure of interest in a coarser scale propagate the results through finer scales until the finest one. Behind such approach there is the hypothesis that at the

lower resolution, small details become less significant relative to the structure of interest.

All these methods and algorithms must be incorporated in a software to provide surgeons with a computational tool for prosthesis design and surgical planning. Modeling and visualization systems have revolutionized many scientific disciplines by providing novel methods to visualize complex data structures and by offering the means to manipulate this data in real-time. With these systems the surgeons are able to: navigate through the anatomy, practice both established and new procedures, learn how to use new surgical tools, and assess their progress. Digital prosthesis design is a beautiful example of such application. Softwares like Mimics, InVesalius and 3-Matic [5], [1] incorporate image processing and geometric design methods to allow medical teams to customize the prosthesis before the implant. In this context, user interaction becomes an important issue.

The techniques mentioned above are applied with the hope to automatically get the geometric model of the prosthesis. However, only the medical team knows exactly the requirements for each specific case. Besides, our algorithms always have a percentage of error. Therefore, we must provide user interactions strategies to the surgeons. We can use traditional mouse-controlled interaction with the three-dimensional model on a standard graphics monitor or virtual reality devices combined with stereo viewing. We can also provide the user with an intuitive sculpting system such that user should be able to freely deform, add and remove material in the digital model.

This chapter covers these issues and is organized as follows. We review related works in section 2. Sections 3 presents the background methods. In section 4 we describe the proposed models to address the mentioned issues. Along this section experimental results are also presented. Next, in section 6 we offer a discussion by presenting some perspectives and issues related to the models described on previous sections. Finally, we offer the conclusions of our work (section 7).

2 Related Works

In older surgical procedures, cranioplasty implants were manufactured directly in the operating-theater where the surgeon modeled the suitable material by hand, namely polymethylmethacrylate (PMMA), to build the prosthesis. The PMMA takes a few minutes to acquire shape and become ready to be implanted in the patient. But, in order for this procedure to be successful, the surgeon needs to have splendid manual ability and still there is an increased chance of infection and rejection [15].

In early 1980, the introduction of 3-D CT scanners and surface reconstruction methods provided a new effective tool for surgical planning. The possibility of generating a geometric (digital) model of the lost part combined with stereolithography rapid prototyping technologies (RP) can make skull prostheses manufacture more accurate and customized for each case [25].

Besides image acquisition, the avenue in this area includes material science, algorithms and computational systems. To be ideal for cranioplasty, the material must have properties like capable of growth, resistant to infection, noncorrosive, stable, inert, among many others (see for a recent study [18]). Up to now, no currently available materials satisfy all the requirements and the research of new materials still remains an active area.

The primary objective of the algorithms in this field is the virtual rebuild of the skull imperfection. It has been performed using (whenever possible) a reflection of the cranium as a guide to fill the holes or by drawing new pixels manually [15].

In order to evaluate the 3D morphology of the bone, thresholding methods were used to exclude the soft tissues. Afterwards, marching methods may be used to acquire the bone's polygonal representation. From the marching result, the creation of the digital model of the implant may begin. This model may also be a polygonal surface or a parametric one (NURBS), according to the fabrication method applied. [26, 10] Mimics, 3-Matic and the free software In Vesalius (Figure 1) are examples of computational systems which are accessible and render these properties [3], [?], [5].

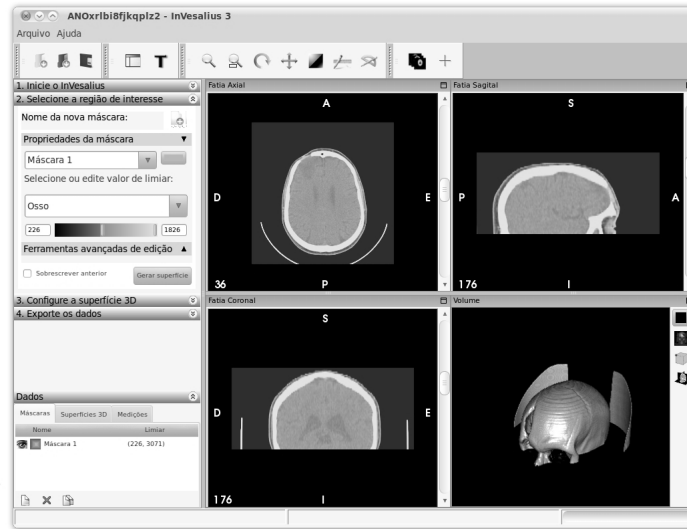


Fig. 1. InVesalius software and surface reconstruction from CT images.

Despite of the simplicity of these functionalities, the above software, have been applied in various surgical planning cases found in the medical literature [15]. This demonstrates that the development of innovative algorithms is a promising research area.

For instance, let us see the method presented in [49], which, to our knowledge, is a state-of-the-art method for digital prosthesis design. At first, user chooses a segment of the image to preserve memory space. Afterwards, each voxel is separated into 5 tetrahedral parts and the marching cubes method is used to acquire a polygonal picture of the imperfect cranium (see Figure 2). Obeying the nomenclature in [49], the respective model has the internal, external and hole surfaces (Figure 2). Next, the boundary link is removed, meaning, the pathway made of edges which is shared by the internal (external) and hole surfaces. In [49] the authors affirm that the angle of the two triangles along the boundary link is smaller– this is the dihedral angle criterion which the algorithms are based upon. Only the edges which meet the three rules obey this criterion:

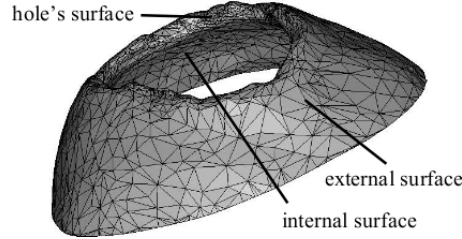


Fig. 2. Defective skull geometry recovered by Tetra-Cubes (Reprinted from [49]).

1. Two vertices of edge are inside the region link.
2. The direction of the edge is closest to the direction of a guiding vector, defined through two guiding points specified by the user.
3. The dihedral angle on the edge is smallest.

Two sets of guiding points must be selected, one for the outside and another for the inside boundary link, to guide the searching algorithm (details in [49]). The result, shown in Figure 3, is used to extract the hole's surface. Now, each boundary link is filled by triangulating [9] the corresponding 3D polygons. So, let a polygon $P = (v_0, v_1, \dots, v_{n-1}, v_n = v_0)$, with $v_i \in V = \mathbb{R}^3$, $i = 0, 1, 2, \dots, n$, and a weight function $\Omega : V^3 \rightarrow L$, where L is a weight set and Ω assigns a weight (area, for instance) to each triangle whose vertices are in P . For $0 \leq i < j \leq n-1$, let $W_{i,j}$ be the weight of the minimum-weight triangulation of the sub-polygon (v_i, v_j) , computed as follows:

1. For $i = 0, 1, \dots, n-2$, let $W_{i,i+1} \leftarrow 0$. For $i = 0, 1, \dots, n-3$, let $W_{i,i+2} \leftarrow \Omega(v_i, v_{i+1}, v_{i+2})$. Set $j \leftarrow 2$.
2. Put $j \leftarrow j + 1$. For $i = 0, 1, \dots, n-j-1$ and $k \leftarrow i + j$, compute:

$$W_{i,k} \leftarrow \min_{i < m < k} [W_{i,m} + W_{m,k} + \Omega(v_i, v_m, v_k)] \quad (1)$$

Let $O_{i,k}$ be the index m where the minimum is achieved.

3. If $j < n - 1$, go back to step 2. Otherwise, $W_{0,n-1}$ holds the weight of the minimum-weight triangulation.
4. Let $S \leftarrow \emptyset$. Call the recursive function *Trace* with parameters $(0, n - 1)$.
Function *Trace*(i, k)
if $i + 2 = k$ then
 $S \leftarrow S \cup \Delta v_i v_{i+1} v_k$
else
Let $o \leftarrow O_{i,k}$
if $o \neq i + 1$ then *Trace*(i, o)
 $S \leftarrow S \cup \Delta v_i v_o v_k$
if $o \neq k - 1$ then *Trace*(o, k)
endif

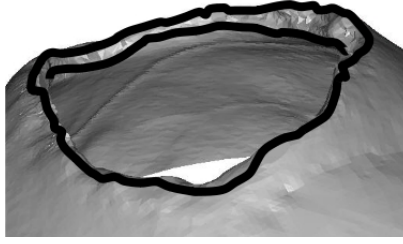


Fig. 3. External and internal boundary links (Reprinted from [49])

In Figure 5 the result is demonstrated after fulfilling the outside boundary link with the recorded triangles. This triangulation must be made better, by adding new vertices and performing edge relaxation in order to acquire a Delaunay-like triangulation (Figure 4) [34]. Fundamentally, given two triangles T_1 and T_2 , along side the edge e , we compute the circum-sphere of the triangle T_1 and see if the vertex $v \in T_2$, on the other edge side, is inside this circum-sphere. The edge is switched in this case. The algorithm also uses a density control factor α as a stopping criterion.

Conclusively, this smoothing procedure is used. Let $\omega : V^2 \rightarrow \Re$ be a weighting function defined on the surface edges. Also, define the weighted umbrella-operator $U_\omega : V \rightarrow E$, from V to the Euclidean space E :

$$U_\omega(v) = -v + \left[\frac{1}{\sum_i \omega(v, v_i)} \right] \sum_{v_i \in B} \omega(v, v_i) v_i, \quad (2)$$

where B is the set of direct neighbors of v . So, given a vertex v , we compute expression (2) and replace v with $v + U_\omega(v)$ to get a smoother version of the surface. Given an edge (v_i, v_j) , the weights ω can be computed by the expression:

$$\omega(v_i, v_j) = \cot(\angle(v_i, v_{k1}, v_j)) + \cot(\angle(v_i, v_{k2}, v_j)),$$

where (v_i, v_{k1}, v_j) and (v_i, v_{k2}, v_j) are the two adjacent triangles of (v_i, v_j) .

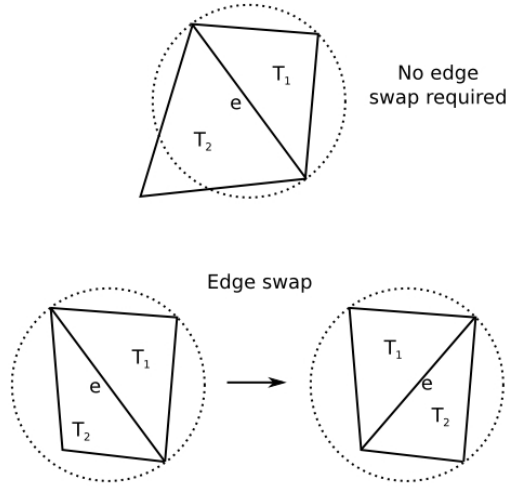


Fig. 4. Process of edge relaxation (Reprinted from [49]).

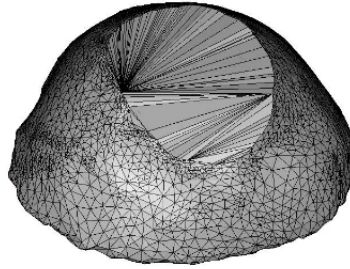


Fig. 5. Triangulated hole (Reprinted from [49]).

3 Background

The field of digital prosthesis modeling includes methods for segmentation, surface reconstruction and computational geometry. Also, multiscale and geometric modeling methods can be used. Besides, user interaction must be considered. So, we present the background in the area by reviewing methods in isosurface extraction, deformable models, wavelets and subdivision surfaces. These methods are the background for the next sections.

3.1 Mass-Spring System and T-Surfaces

In the mass-spring system the surface nodes work as masses and the edges define the linear springs with damping. So, given a particle i with mass m_i and position

vector x_i , the force system is composed by the elastic ($f_{elastic}^i$), gravitational (f_{grav}^i) and damping (f_{damp}^i) forces, defined respectively, by [45, 20]:

$$\mathbf{f}_{elastic}^i = \sum_{j \in V} c_{ij} (l_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|) \frac{(\mathbf{x}_i - \mathbf{x}_j)}{\|\mathbf{x}_i - \mathbf{x}_j\|}, \quad (3)$$

where V is the set of nodes linked to \mathbf{x}_i , c_{ij} is the stiffness of the spring linking the nodes \mathbf{x}_i and \mathbf{x}_j and l_{ij} the spring rest length;

$$\mathbf{f}_{grav}^i = m_i \mathbf{g}, \quad (4)$$

$$\mathbf{f}_{damp}^i = \gamma_i \dot{\mathbf{x}}_i, \quad (5)$$

\mathbf{g} is the gravity field and γ_i is the damping factor. Following Newton's Laws, we get the following evolution equation:

$$m_i \ddot{\mathbf{x}}_i = \mathbf{f}_{elastic}^i + \mathbf{f}_{damp}^i + \mathbf{f}_{grav}^i, \quad (6)$$

In this development there is a one-to-one relationship between the mesh topology and the springs. We can relax such coupling by allowing the interaction between particles that are not connected by any edge. Following this idea, we can include shearing and bending besides material tension without including extra edges. The Figure 6 represents this idea by showing a regular mesh, in which the particles are connected by structural springs to account for tension, as well as additional springs that do not correspond to edges. They are diagonal springs for shearing, and interleaving springs for bending [45]. Each spring is governed by Hook's Law and all the corresponding forces can be computed by an expressions analogous to (3)- (5).

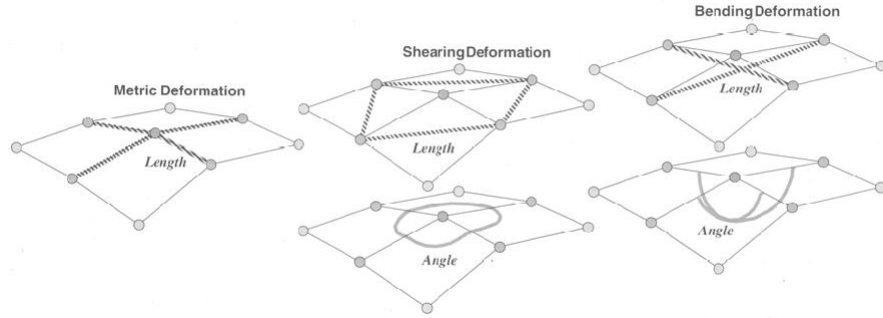


Fig. 6. Square mesh with additional links to model bending and shearing forces. There are no edge supporting the additional links (Reprinted from [45]).

The T-Surfaces approach is a topologically adaptable deformable model which is composed basically by three components [31]: (1) a tetrahedral decomposition (CF-Triangulation) of the image domain $D \subset \mathbb{R}^3$; (2) a mass-spring

system; (3) a *Characteristic Function* χ defined on the grid nodes which distinguishes the interior ($Int(S)$) from the exterior ($Ext(S)$) of a surface S :

$$\chi : D \subset \mathbb{R}^3 \rightarrow \{0, 1\} \quad (7)$$

where $\chi(p) = 1$ if $p \in Int(S)$ and $\chi(p) = 0$, otherwise, where p is a node of the grid.

Following the classical nomenclature, a vertex of a tetrahedron is called a *node* and the collection of nodes and triangle edges is called the grid Γ_s . A tetrahedron (also called a simplex) σ is a *transverse* one if the characteristic function χ in equation (7) changes its value in σ . Analogously, for an edge.

In this framework, the reparameterization of a surface is done by [31]: (1) Taking the intersections points of the surface with the grid; (2) Find the set of transverse tetrahedra (*Combinatorial Manifold*); (3) For each transverse edge choose an intersection point belonging to it; (4) Connect these points properly.

In this reparameterization process, the transverse simplices play a central role. Given such a simplex, we choose in each transverse edge an intersection point to generate the new surface patch. In general, we will have three or four transverse edges in each transverse tetrahedron (Figure 8). The former gives a triangular patch and the later defines two triangles. So, at the end of the step (4) we have a triangular mesh. Each triangle is called a *triangular element* [31].

As an example for 2D, consider the characteristic functions (χ_1 and χ_2) relative to the two contours pictured in Figure 7. The functions are defined on the vertices of a CF-triangulation of the plane. The vertices marked are those where $\max\{\chi_1, \chi_2\} = 1$. Observe that they are enclosed by a merge of the contours. This merge can be approximated by a curve belonging to the region obtained by tracing the transverse triangles. The same would be true for more than two contours (and obviously for only one).

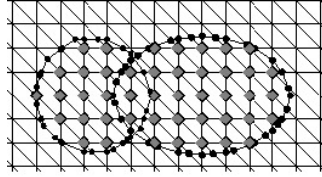


Fig. 7. Two snakes colliding with the inside grid nodes and snaxels marked.

Besides the forces (3) and (5), the model also has a normal force which can be weight as follows [31]:

$$F_i = k(sign_i)n_i, \quad (8)$$

where n_i is the normal vector at node i , k is a scale factor, and $sign_i = +1$ if $I(v_i) > T$ and $sign_i = -1$ otherwise (T is a threshold for image I). This force is used to push the model towards image edges until it is opposed by external image forces.

The forces given in expressions (3) and (8) are internal forces. The external force is defined as a function of the image data, according to the features we seek. One possibility is:

$$Image :: Force :: f_i^t = -\gamma_i \nabla P; \quad P = \|\nabla I\|^2. \quad (9)$$

The evolution of the surface is governed by the following dynamical system:

$$v_i^{(t+\Delta t)} = v_i^t + h_i \left(\vec{\alpha}_i^t + \vec{F}_i^t + \vec{f}_i^t \right), \quad (10)$$

where h_i is an evolution step. During the T-Surfaces evolution some grid nodes become interior to a surface. Such nodes are called *burnt nodes* and its identification is fundamental to update the characteristic function [31]. To deal with self-intersections of the surface the T-Surfaces model incorporates an entropy condition: *once a node is burnt it stays burnt*. A termination condition is obtained based on the number of deformations steps that a simplex has remained a transverse one.

3.2 Shape Model

We consider an object to be described by points, referred to as landmark points. These points may represent the boundary of the object or some structures of interest in the images of a database.

In general, the landmark points are (manually) determined in a set of s training images. From these collections of landmark points, a point distribution model [44] is constructed as follows. The landmark points are stacked in shape vectors:

$$\mathbf{x}^i = (x_1^i, y_1^i, x_2^i, y_2^i, \dots, x_n^i, y_n^i)^T, \quad i = 1, 2, \dots, s, \quad (11)$$

where (x_j^i, y_j^i) refers to the landmark j for the image i .

Principal component analysis (PCA), or KL-Transform [44, 21, 17], is applied to the shape vectors by computing the mean shape:

$$\bar{\mathbf{x}} = \frac{1}{s} \sum_{i=1}^s \mathbf{x}^i, \quad (12)$$

the covariance matrix:

$$\mathbf{S} = \frac{1}{s-1} \sum_{i=1}^s (\mathbf{x}^i - \bar{\mathbf{x}}) (\mathbf{x}^i - \bar{\mathbf{x}})^T, \quad (13)$$

and the eigenvectors of the covariance matrix sorted in decreasing order of the corresponding eigenvalues:

$$\{\phi_1, \phi_2, \dots, \phi_{2n}\}. \quad (14)$$

The eigenvectors corresponding to the largest t eigenvalues λ_i are retained in a matrix Φ . A shape \mathbf{x} can be now approximated by:

$$\min_{\mathbf{b}} \|\mathbf{x} - (\bar{\mathbf{x}} + \Phi \cdot \mathbf{b})\|, \quad (15)$$

where \mathbf{b} is a vector of t elements containing the model parameters.

3.3 Surface Reconstruction

Isosurface extraction is one of the most used techniques for surface reconstruction in 3D data sets. Depending on the data type (time-varying or stationary) and the data size, many works have been done to improve the basic methods in this area [43]. In this paper we consider two kinds of isosurface generation methods: the marching ones and continuation ones.

In Marching Cubes, each surface-finding phase visits all cells of the volume, normally by varying coordinate values in a triple "for" loop [30]. As each cell that intersects the isosurface is found, the necessary polygon(s) to represent the portion of the isosurface within the cell is generated. There is no attempt to trace the surface into neighboring cells. Space subdivision schemes (like Octree and k-d-tree) have been used to avoid the computational cost of visiting cells that the surface do not cut [13, 43].

Once the T-Surfaces grid is a CF one, the Tetra-Cubes is a natural choice [11]. Like in the marching cubes, its search is linear: each cell of the volume is visited and its simplices (tetrahedron) are searched to find surfaces patches. Following marching cubes, its implementation uses auxiliary structures based on the fact that the topology of the intersections between a plane and a tetrahedron can be reduced to three basic configurations pictured on Figure 8.

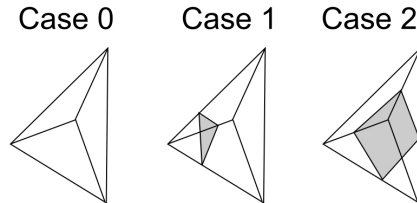


Fig. 8. Basic types of intersections between a plane and a simplex in 3D.

Unlike marching methods, continuation algorithms attempt to trace the surface into neighboring simplices [8]. Thus, given a transverse simplex, the algorithm searches its neighbors to continue the surface reconstruction. The key idea

is to generate the combinatorial manifold (set of transverse simplices) that holds the isosurface.

The following definition will be useful. Let's suppose two simplices σ_0, σ_1 , which have a common face and the vertices, $v \in \sigma_0$ and $v' \in \sigma_1$ both opposite the common face. The process of obtain v' from v is called *pivoting*. Let us present the basic continuation algorithm [8].

PL Generation Algorithm:

```

Find a transverse triangle  $\sigma_0$ ;
 $\Sigma = \{\sigma_0\}$ ;  $V(\sigma) = \text{set of vertices of } \sigma$ ;
while  $V(\sigma) \neq 0$  for some  $\sigma \in \Sigma$ 
.   get  $\sigma \in \Sigma$  such that  $V(\sigma) \neq 0$ ;
.   get  $v \in V(\sigma)$ ;
.   obtain  $\sigma'$  from  $\sigma$  by pivoting  $v$  into  $v'$ 
.   if  $\sigma'$  is not transverse
.       then drop  $v$  from  $V(\sigma)$ ;
.   else
.       if  $\sigma' \in \Sigma$  then
.           drop  $v$  from  $V(\sigma)$ ,  $v'$  from  $V(\sigma')$ 
.       else
.            $\Sigma \Leftarrow \Sigma + \sigma'$ ;
.            $V(\sigma') \Leftarrow \text{set of vertices of } \sigma'$ ;
.           drop  $v$  from  $V(\sigma)$ ,  $v'$  from  $V(\sigma')$ 

```

Differently from Tetra-Cubes, once started the generation of a component, the algorithm runs until it is completed. However, the algorithm needs a set of seed simplices to be able to generate all the components of an isosurface. This is an important point when comparing continuation and marching methods.

If we do not have guesses about seeds, every simplex should be visited. Thus, the computational complexity of both methods are the same ($O(N)$ where N is the number of simplices). However, if we know in advance that the target boundary is connected we do not need to search for inside components. Thus, the computational cost is reduced if compared with Tetra-Cubes. That is way we use continuation methods in [16, 40] to get the initial surfaces.

3.4 Wavelet Theory and Multiscale Analysis

Wavelets can be introduced in the context of basis functions and scale-varying basis functions. Historically, such theory received a great impact in the 1930s, when several groups working independently had established some foundations of such theory and its applications. For instance, by using the scale-varying Haar basis function, the physicist Paul Levy, investigated at that time the Brownian motion. He found the Haar basis function superior to the Fourier basis functions for that study [19]. In 1980, Grossman and Morlet, a physicist and an engineer, broadly defined wavelets in the context of quantum physics. In 1985, Stephane Mallat discovered some relationships between quadrature mirror filters, pyramid algorithms, and orthonormal wavelet bases. A couple of years later, Ingrid

Daubechies used Mallat's work to perform perhaps the most elegant development in this field. The set of wavelet orthonormal basis functions constructed has become a very remarkable work for wavelet applications since then [19].

Definition 5: A multiscale analysis (MSA) of $L^2(\mathbb{R})$ is an increasing sequence of closed subspaces, called **scale spaces**, $V_m \subset L^2(\mathbb{R})$:

$$\{0\} \subset \cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots \subset L^2(\mathbb{R}) \quad (16)$$

such that the following are true:

$$\overline{\bigcup_{m \in \mathbb{Z}} V_m} = L^2(\mathbb{R}), \quad (17)$$

$$\bigcap_{m \in \mathbb{Z}} V_m = \{0\}, \quad (18)$$

$$f(x) \in V_m \iff f(2^m x) \in V_0 \quad (19)$$

There is a function $\varphi \in L^2(\mathbb{R})$ whose interger translates generate a Riesz basis of V_0 (that mean, V_0 is the closure of the set $\text{span}\{\varphi_{m,k} \mid k \in \mathbb{Z}\}$):

$$V_0 = \overline{\text{span}\{\varphi(x-k) \mid k \in \mathbb{Z}\}}$$

and

$$A \sum_{k \in \mathbb{Z}} c_k^2 \leq \left\| \sum_{k \in \mathbb{Z}} c_k \varphi(\cdot - k) \right\|_{L^2}^2 \leq B \sum_{k \in \mathbb{Z}} c_k^2 \quad (20)$$

for all $\{c_k\}_{k \in \mathbb{Z}} \in l^2(\mathbb{Z})$. A and B are positive constants.

It is important to emphasize the following remarks:

(a) Conditions (17) and (18) are satisfied by many families $\{V_m\}_{m \in \mathbb{Z}}$. Property (19) is the special feature of an MSA: the spaces V_m are scaled versions of the basic space V_0 , which is spanned by the translations of φ , the scaling function.

For $m \rightarrow \infty$ the functions in V_m are dilated, i.e. their details enlarged. If m tends to $-\infty$, then the spaces V_m contain smaller and smaller structures. The limits

$$\lim_{m \rightarrow +\infty} \|P_m f\|_{L^2} = 0, \quad (21)$$

$$\lim_{m \rightarrow -\infty} \|P_m f - f\|_{L^2} = 0, \quad (22)$$

make this interpretation precise. P_m denotes the orthogonal projector onto V_m . The following phrase has been adopted, but it is more suggestive than exact:

$P_m f$ is the representation of f on the ‘scale’ V_m and contains all details of f up to the size 2^m .

(b) The relation (20) implies V_0 is invariant under interge translation

$$f \in V_0 \iff f(x - k) \in V_0 \quad \text{for } k \in Z. \quad (23)$$

With (19) it follows that

$$f \in V_m \iff f(x - 2^m k) \in V_m \quad \text{for } k \in Z. \quad (24)$$

(c) The space V_m is spanned by the functions:

$$\varphi_{m,k}(x) := 2^{-m/2} \varphi(2^{-m}x - k), \quad (25)$$

$$V_m = \overline{\text{span} \{ \varphi_{m,k} \mid k \in Z \}}. \quad (26)$$

This is based on (19) and (20). The functions in (25) all have the same L^2 -norm $\|\varphi_{m,k}\|_{L^2} = \|\varphi\|_{L^2}$.

Scaling Equation: The scaling function φ satisfies a scaling equation, i.e. there is a sequence $\{h_k\}_{k \in Z}$ of real numbers such that:

$$\varphi(x) = \sqrt[2]{2} \sum_{k \in Z} h_k \varphi(2x - k). \quad (27)$$

The key to the construction of both orthogonal wavelet bases (wavelet spaces) and fast algorithms lies in equation (27). In this way, we denote by W_m the orthogonal complement of V_m in V_{m-1} , that means:

$$V_{m-1} = W_m \oplus V_m, \quad V_m \perp W_m. \quad (28)$$

If Q_m is the orthogonal projector of $L^2(\mathbb{R})$ in W_m and P_{m-1} denotes the orthogonal projector onto V_{m-1} (likewise in expression (21)-(22)), then, decomposition (28) means:

$$P_{m-1} = Q_m + P_m. \quad (29)$$

From (28) It is clear that:

$$V_{m-1} = W_m \oplus (W_{m+1} \oplus V_{m+1}), \quad V_{m+1} \perp W_{m+1}, \quad (30)$$

and so on. Therefore, using the MSA definition, it follows that:

$$V_{m-1} = \bigoplus_{j \geq m} W_j \quad (31)$$

and so:

$$L^2(\mathbb{R}) = \bigoplus_{j \in Z} W_j. \quad (32)$$

The spaces W_m , with $m \in Z$, are named **wavelet spaces**.

Let $\{V_m\}_{m \in Z}$ be an MSA generated by the orthogonal scaling function $\varphi \in V_0$ and the function $\psi \in V_{-1}$, defined by

$$\psi(x) = \sqrt{2} \sum_{k \in Z} g_k \varphi(2x - k) = \sum_{k \in Z} g_k \varphi_{-1,k}(x), \quad (33)$$

$$g_k = (-1)^k h_{1-k}, \quad (34)$$

where $\{h_k\}_{k \in Z}$ are the coefficients of the scaling equation (27).

Theorem 1: The function ψ defined by expressions (33)-(34) has the following properties:

- (i) $\{\psi_{m,k}(\cdot) = 2^{-m/2} \psi(2^{-m}x - k) \mid k \in Z\}$ is an orthonormal basis for W_m ,
- (ii) $\{\psi_{m,k} \mid m, k \in Z\}$ is an orthonormal basis for $L^2(\mathbb{R})$,
- (iii) ψ is a wavelet; that means: $c_\psi = 2\pi \int_{\mathbb{R}} |\omega|^{-1} \left| \hat{\psi}(w) \right|^2 dw = 2 \ln 2 < \infty$

In section 5.1 we will discuss the application of this approach do help the definition of the boundary of the hole pictured in Figure 3.

3.5 2D Deformable Model Balloon

Deformable models are very useful for shape recovery in 2D/3D images. In our case we will apply a 2D deformable model, a balloon-like one [14]. Geometrically, this model is described by a parametric contour c embedded in a domain $D \subset \mathbb{R}^2$:

$$c : [0, 1] \rightarrow D \subset \mathbb{R}^2,$$

$$c(s) = (x(s), y(s)). \quad (35)$$

From the dynamic viewpoint, we have a deformable contour which is viewed as a time-varying curve

$$c(s, t) = (x(s, t), y(s, t)). \quad (36)$$

In this formulation, the Lagrange equations of motion can be expressed as:

$$\dot{c} - (\omega_1 c')' + (\omega_2 c'')'' = \mathbf{F}_{ext} + k\mathbf{n}(c), \quad (37)$$

subject to the specification of the following boundary conditions:

$$c(0, t), c(1, t), c'(0, t), c'(1, t), \quad (38)$$

where $\dot{c} \equiv \frac{\partial c}{\partial t}$, $c' \equiv \frac{\partial c}{\partial s}$, $c'' \equiv \frac{\partial^2 c}{\partial s^2}$, $\mathbf{n}(c)$ is the unit normal over the curve and k is a scale factor. The field \mathbf{F}_{ext} means an external force, which depends on image elements or constraints.

In our case, the curve is an open one, fixed in the end points of the lesion following the tangents at that points. Besides these boundary conditions, we must set the initial curve to complete the initialization of the balloon.

So, in order to include the initial curve in the numerical method, we must give a sequence of points $C^0 = c(i\Delta s, 0) = c_i^0$, com $i = 0, \dots, N$, which is the discrete version of the initial curve. Such curve must satisfies the boundary conditions also. Therefore, using a reference frame as pictured on Figure 9, which has both the end points on the horizontal axis (one in the origin and the other one at $x = L$) we can write the constraints as:

$$\begin{aligned} C(0) &= 0, \\ C(L) &= 0, \\ C'(0) &= \tan \theta_0, \\ C'(L) &= \tan \theta_1. \end{aligned} \tag{39}$$

Besides, the initial shape should be as closer as possible to the target to reduce time computation. We observe that a cubic curve, represented by $C(x) = ax^3 + bx^2 + cx + d$ fits all of these requirements. With a simple algebra we can demonstrate that:

$$a = \frac{\tan \theta_0 + \tan \theta_1}{L^2} \tag{40}$$

$$b = -\frac{2 \tan \theta_0 + \tan \theta_1}{L} \tag{41}$$

$$c = \tan \theta_0 \tag{42}$$

$$d = 0 \tag{43}$$

By considering that $s \in [0, 1]$ in the balloon definition, we can represent the initial curve as

$$\begin{aligned} c(s) &= (Ls, L(\tan \theta_0 + \tan \theta_1)s^3 - \\ &\quad L(2 \tan \theta_0 + \tan \theta_1)s^2 + L \tan \theta_0 s). \end{aligned} \tag{44}$$

We can approximate the derivatives in the expression (37) through finite differences, with discretization steps of $\Delta s = 1/N$ and $\Delta t = 1/M$ for space and time, respectively and by considering ω_1, ω_2 constants:

$$\dot{c} \approx \frac{c_i^\tau - c_i^{\tau-1}}{\Delta t},$$

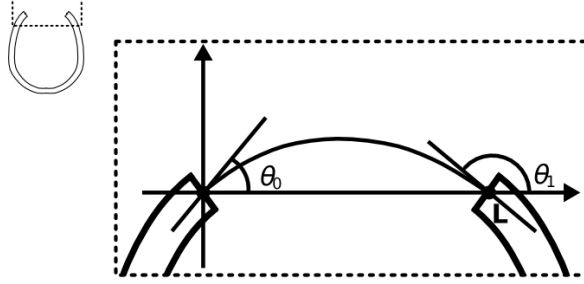


Fig. 9. Reference frame for balloon initialization.

$$c'' \approx \frac{1}{\Delta s^2} [c_{i-1}^\tau - 2c_i^\tau + c_{i+1}^\tau],$$

$$c''' \approx \frac{1}{\Delta s^4} [c_{i-2}^\tau - 4c_{i-1}^\tau + 6c_i^\tau - 4c_{i+1}^\tau + c_{i+2}^\tau],$$

where $c_i^\tau = c(i\Delta s, \tau\Delta t)$, with $i = 0, \dots, N$ and $\tau = 0, \dots, M$.

We shall observe that, once the *snake* is an open curve with fixed endpoints (say, $P_0 \in P_N$), then $c_0^\tau = P_0$ and $c_N^\tau = P_N$, for any τ . Also, by using the expressions:

$$c'(0, t) \approx \frac{c_1^\tau - c_0^\tau}{\Delta s}, \quad (45)$$

$$c'(1, t) \approx \frac{c_N^\tau - c_{N-1}^\tau}{\Delta s}, \quad (46)$$

and considering that $c'(0, t) = \mathbf{u}_0$ and $c'(1, t) = \mathbf{u}_1$, we get $c_1^\tau = P_0 + \Delta s \mathbf{u}_0$ and $c_{N-1}^\tau = P_N - \Delta s \mathbf{u}_1$ for any τ . Therefore, the points c_0^τ , c_1^τ , c_{N-1}^τ and c_N^τ are known along the whole balloon evolution.

From this fact and substituting the derivative expressions in equation (37), we get [14]:

$$(I + \Delta t A) C^\tau = C^{\tau-1} + \Delta t (kF - L), \quad (47)$$

where I is the identity matrix of order $(N-3)$, $C^\tau = [c_2^\tau \dots c_{N-2}^\tau]^T$,

$$F = [\mathbf{n}(c_2^{\tau-1}) \dots \mathbf{n}(c_{N-2}^{\tau-1})]^T,$$

$$L = [l_1 \ l_2 \ 0 \ \dots \ 0 \ l_3 \ l_4]^T,$$

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & & & & \\ a_1 & a_0 & a_1 & a_2 & & & \\ a_2 & a_1 & a_0 & a_1 & a_2 & & \\ & a_2 & a_1 & a_0 & a_1 & a_2 & \\ & & & \ddots & & & \\ & & & & a_2 & a_1 & a_0 & a_1 & a_2 \\ & & & & & a_2 & a_1 & a_0 & a_1 & a_2 \\ & & & & & & a_2 & a_1 & a_0 & a_1 \\ & & & & & & & a_2 & a_1 & a_0 \end{bmatrix},$$

where:

$$\begin{aligned} a_2 &= \frac{\omega_2}{\Delta s^4} \\ a_1 &= -\frac{\omega_1}{\Delta s^2} - 4\frac{\omega_2}{\Delta s^4} \\ a_0 &= 2\frac{\omega_1}{\Delta s^2} + 6\frac{\omega_2}{\Delta s^4} \\ l_1 &= a_2 P_0 + a_1(P_0 + \Delta s \mathbf{u}_0) \\ l_2 &= a_2(P_0 + \Delta s \mathbf{u}_0) \\ l_3 &= a_2(P_N - \Delta s \mathbf{u}_1) \\ l_4 &= a_2 P_N + a_1(P_N - \Delta s \mathbf{u}_1) \end{aligned}$$

In this way, the system (47) gives a method to compute C^τ once we have $C^{\tau-1}$. Besides, we shall notice that $(I + \Delta t A)$ is a symmetric pentadiagonal banded matrix. So, it is computationally efficient to solve the system through LU decomposition of the matrix A [36]. The balloon comes to rest when the internal and external forces balance, which implies that:

$$\|C^{t+\Delta t} - C^t\| \leq \varepsilon, \quad (48)$$

where $C^{t+\Delta t}$ and C^t are the curves at time $t + \Delta t$ and t , respectively, and ε is a pre-defined parameter.

The method is sensitive to the choice of the coefficients ω_1 and ω_2 . Following [14], we set these parameters such that ω_1 and ω_2 are proportional to Δs^2 and Δs^4 , respectively.

3.6 Subdivision Scheme

In the last decade, subdivision surfaces have found their way into wide applications in geometric modeling and animation. One reasons for this fact is that subdivision is intricately linked to multiresolution. Besides, constructing surfaces through subdivision elegantly addresses issues of covering arbitrary topology, level-of-detail modeling, numerical requirements for further application in finite element, elegant formulation through mathematical tools such as wavelets, etc.

For instance, let us consider the Sabin's algorithm [37] which is a variant of the traditional Catmull-Clark's method [12]. The input for Sabin's algorithm is a closed net $N = (V, E)$, where V is the set of vertices connected according to a topology E (a net is closed if each edge is shared by exactly two faces). Then, the subdivision algorithm will compute a new net $N' = (V', E')$ by applying the following rules over the input net N (see [37] for details):

1. For each old face f , make a new face-vertex v_f^* as the weighted average of the old vertices of f , with weights W_n that depend on the valency n of each vertex.
2. For each old edge e , make a new edge-vertex v_e^* as the weighted average of the old vertices of e and the new face vertices associated with the two faces originally sharing e . The weights W_n (which are the same as the used in rule 1) depend on the valency n of each vertex (the valency of new vertices is equal to the one of the corresponding faces).
3. For each old vertex v , make a new vertex-vertex v_v^* at the point given by the following linear combination, whose coefficients α_n , β_n and γ_n depend on the valency n of v :
 $\alpha_n \cdot (\text{the centroid of the new edge vertices of the edges meeting at } v) + \beta_n \cdot (\text{the centroid of the new face vertices of the faces sharing those edges}) + \gamma_n \cdot v.$

The topology E' of the new net is calculated by the following rule:

For each old face f and for each vertex v of f , make a new quadrilateral face whose edges join v_f^* and v_v^* to the edge vertices of the edges of f sharing v (see figure 10).

The weights W_n are computed as follows. Let $n > 2$ be the valency of a vertex and $k = \cos(\pi/n)$. Compute the real roots of $x^3 + (4k^2 - 3)x - 2k = 0$ and take the only one satisfying $x > 1$. So, the weights are calculated by:

$$\begin{aligned} W_n &= x^2 + 2kx - 3 \\ \alpha_n &= 1 \\ \gamma_n &= \frac{kx + 2k^2 - 1}{x^2(kx + 1)} \\ \beta_n &= -\gamma_n \end{aligned} \tag{49}$$

In the context of polygonal surfaces, digital prosthesis design can be cast in the problem of filling n -sided holes in the surface. Therefore, algorithms that generates a subdivision surface which connects smoothly with the surface around the hole can be useful for prosthesis modeling.

4 Computer-Aided Design Methods and Systems

In this section, we discuss some proposals for prosthesis modeling. We start with a balloon model that is applied slice-by-slice for reconstruction of the lost part

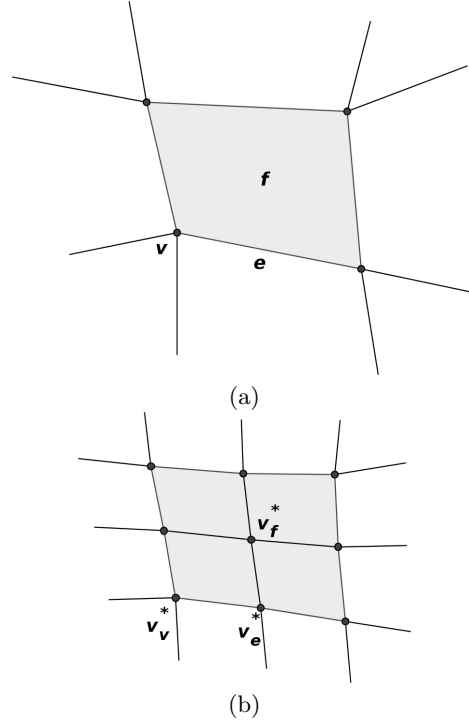


Fig. 10. (a) Initial net and highlighted face (Reprinted from [27]). (b) New net obtained with the Sabin's algorithm (Reprinted from [27]).

(section 4.1). Then, a marching cubes technique is used to get the final geometry. On the other hand, we can perform all the tasks in 3D space. The T-Surfaces plus isosurface generation models offer a general framework for surface reconstruction and user interaction [41]. Therefore, we can explore this framework for digital prosthesis design, as discussed in section 5. The wavelet theory can be applied to simplify the detection of the boundary points of the hole. Besides subdivision surfaces have been used for filling holes methods. So, we discuss the application of these methods for prosthesis modeling in section 5.1. We describe also experimental results. In section 5.2, we present a computational tool that we are developing for image processing and visualization which can be used for digital prosthesis design.

4.1 Balloon Based Model

In this section, we propose a new methodology for digital prosthesis generation, which can be roughly divided into 5 stages:

1. Segmentation: Extract the bone from the other tissues

2. Feature Extraction: get geometric information about the frontier of the lesion (terminal points and their tangents).
3. Deformable Model: With the information from step 2, find the appropriate curve, using a deformable model of type *balloon*.
4. Lesion Reconstruction: Each patch obtained in the step above is dilated to complete the prosthesis volume.
5. Surface Reconstruction: Marching cubes is applied to generate the digital version of the prosthesis geometry.

To accomplish the first step, we apply a simple thresholding technique based on image inspection. In the actual implementation, the obtained result is manually processed, frame-by-frame, to get the end points and their tangents (second stage). Next, we apply a deformable model, a balloon-like model [14], to compute the patch of the prosthesis in each image frame. We take the end points and tangents computed in the step 2 as boundary conditions for the deformable model. In the final stages, we recover the lost part of the bone, following the balloon result in each frame and, finally, reconstruct the geometry of the prosthesis. We demonstrate the advantages of our technique when compare with the one presented in [49].

The standard format for CT images is DICOM (Digital Imaging and Communications in Medicine) [35]. CT images of the head show bone and soft tissues (brain, skin, etc). In a DICOM image, the data matrix has real values in the range $[-1000, 1000]$, called TC numbers. A simple binarization technique may then be used to bring out just the structure elicited since the TC number for bone is in the $[400, 1000]$ range.

Now, once the bone is segmented, we take each frame and get the end points of the lesion and their tangents (see Figure 9) through an user interaction procedure.

Now, we apply the proposed model to generate the prosthesis for repairing the defective skull pictured on Figure 11.(a). In this case, we also could use a reflection technique, based on the assumption of skull symmetry, to reconstruct the skull. The defective region intersects 36 slices of the whole image volume.

In the experiments of this section we set null the external force in expression (37). Following [14], the parameters ω_0 and ω_1 are set to $(\Delta s)^2$ and $(\Delta s)^4$, where Δs is the discretization step applied to get the numerical solution of equation (37). The value of the normal force scale parameter is $k = -1/16$. The stopping criterium is based on expression (48), with 10^{-3} . The Table 1 reports some statistics about the initialization (expression (44)) and evolution of the balloon model for the 36 slices that intersect the defective region.

Considering that the mean number of interactions of the numerical scheme defined by expression (47) is 123 and we have 21 snaxels for each instance of the balloon model, we can say that the computational cost is not expensive. The Figure 11.(a) shows the defective skull and Figure 11.(b) picture the balloon result for one of the slices. Once the balloon stops evolution, we dilate the result using a 5×5 mask centered in the snaxels. Figures 11.(c)-(e) show three viewpoints of the repaired skull. A visual inspection indicates that the prosthesis geometry

Parameter	Min	Max
θ_0	6°	10°
θ_1	170°	174°
No. Interactions	100	189

Table 1. Statistics for initialization parameters (θ_0, θ_1) and number of interactions of balloon.

suitably reconstruct the defective region. However, the method is sensitive to the parameters choice. So, we must be careful about this point during the setup of the balloon technique.

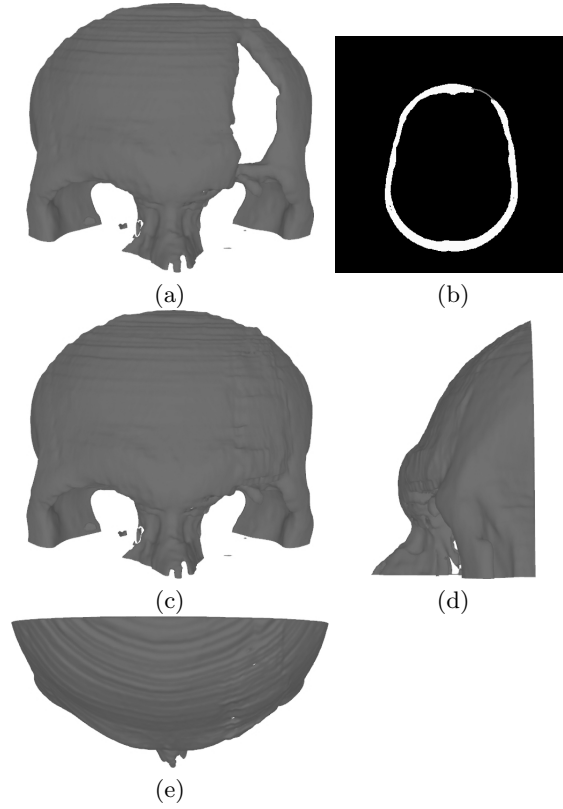


Fig. 11. (a) Original defective skull. (b) Volume slice with the balloon result. (c)-(e) Skull surface reconstructed.

It is worthwhile to compare our method with the one proposed in [49], summarized on section 2. The Figure 12 pictures the solution obtained with that

method. The method proposed in [49] does not take into account the tangent directions at the boundary of the hole. Therefore, the obtained prosthesis may not fit the curvature of the skull. In fact, the method has a bias towards planar shapes due to the fact that there is no any constraint related to local curvature. We can check this problem in the result pictured on Figure 12.

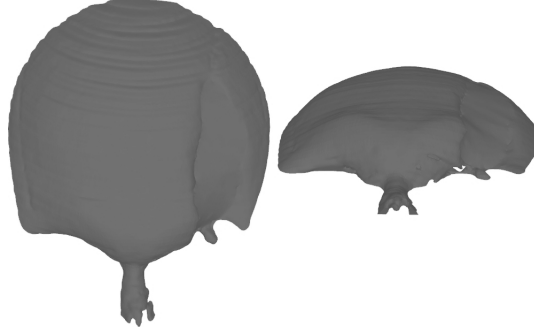


Fig. 12. Viewpoints of the solution obtained by the method described in [49].

We have also some considerations about the algorithm that searches the boundary link, described in section 2. In our implementation of that algorithm, we get suitable results without testing if the vertices are interior to the polygonal frontier (rule 1), as done in [28]. It is important to notice that this polygon is a curve in the $3D$ space which makes the determination of inner points a non-trivial task. Besides, we define an upper bound for the number of triangles generated in the mesh refinement algorithm in order to avoid too many triangles in low curvature regions.

5 Isosurface Methods and T-Surfaces

In this section we describe an approach which integrates the T-Surfaces model and isosurface generation methods in a general framework for surface reconstruction in 3D medical images (see [41] for details). The reparameterization of T-Surfaces gives the link between the above isosurface generation methods and T-Surfaces model. To explain this, let us define an *Object Characteristic Function* as:

$$\begin{aligned} \chi(p) &= 1, & \text{if } I(p) < T, \\ \chi(p) &= 0, & \text{otherwise,} \end{aligned} \tag{50}$$

where p is a node of the triangulation.

If we apply tetra-cubes or continuation method to this field, we get a set of piecewise linear (PL) surfaces that involve the structures of interest. From the

way the PL surfaces are generated, each connected component \widehat{M} so obtained has the following properties: (1) The intersection $\sigma_1 \cap \sigma_2$ of two triangles $\sigma_1, \sigma_2 \in \widehat{M}$ is empty, a common vertex or edge of both triangles; (2) An edge $\tau \in \widehat{M}$ is common to at most two triangles of \widehat{M} ; (3) \widehat{M} is locally finite, that is, any compact subset of \mathbb{R}^3 meets only finitely many cells of \widehat{M} .

A polygonal surface with such property is called a Piecewise Linear Manifold (*PL Manifold*). From the reparameterization process of section 3.1, we can see that a T-Surface is also a PL Manifold. Thus, the isosurface extraction methods can be straightforward used to initialize T-Surfaces with the Object Characteristic Function as the initial Characteristic Function.

But, what kind of isosurface method should be used? Based on the discussion of section 3.3, about tetra-cubes and continuation methods, we can conclude that, if we do not have topological and scale restrictions, marching methods are more appropriated to initialize the T-Surfaces. In this case, it is not worthwhile to attempt to reconstruct the surface into neighboring simplices because all simplices should be visited to find surface patches.

However, for the T-Surfaces reparameterization (steps (1)-(4) of section 3.1), the situation is different. Now, each connected component is evolved at a time. Thus it is interesting a method which generates only the connected component being evolved, that is, the PL Generation algorithm of section 3.3.

The segmentation/surface reconstruction method that we propose in [41] is based on the following steps: (1) Extract region based statistics; (2) Coarser image resolution; (3) Define the Object Characteristic Function; (4) PL manifold extraction by the tetra-cubes; (5) If need, increase the resolution. Return to step (3). (6) Apply T-Surfaces model; (7) User interaction if need.

It is important to highlight that T-Surfaces model can deal naturally with the self-intersections that may happen during the evolution of the surfaces obtained by step (4). This is an important advantage of T-Surfaces.

Among the surfaces extracted in step (4), there may be open surfaces which starts and end in the image frontiers, small surfaces corresponding to artifacts or noise in the background. The former is discarded by a simple automatic inspection. To discard the later, we need a set of pre-defined features (volume, surface area, etc), and corresponding lower bounds. For instance, it is straightforward to set the volume lower bound as $8(r)^3$, where r is the dimension of the grid cells. Besides, some polygonal surfaces may contain more than one object of interest. Now, we can use upper bounds for the features. These upper bounds are application dependent (anatomical elements can be used).

The surfaces whose interior have volumes larger than the upper bound will be processed in a finer resolution. When the grid resolution of T-Surfaces is increased we just reparameterize the model through the finer grid and evolve the corresponding T-Surfaces.

Besides, due to inhomogeneities of the image field (supposed gray level), some objects may be split in the step (4). Sometimes, T-Surfaces model can not merge them again. To correct these problems, the user can manually burn some grid nodes to force merges or splits. From the entropy condition, these nodes remain

burnt until the end of the process. This functionality can be implemented by selecting grid nodes with a pointer (mouse, for example), through implicitly defined surfaces, or through a *virtual* scalpel.

In order to demonstrate the potential of the whole framework let us consider the following example. We have three spheres of intensity 50 placed in a noisy $150 \times 150 \times 150$ image volume. The spheres were previously extracted by the proposed method with the following parameters: grid $5 \times 5 \times 5$, freezing point = 10, $\gamma = 0.01$, $k = 1.222$; $c = 0.750$. Every sphere has radius 30 (pixels).

In this case, the merge is forced through an implicit defined surface placed between the spheres (Figure 13.a). Grid nodes inside the surface are easily detected by its equation and then burnt. During the evolution, the 4 surfaces will merge and the final result is a connected surface (Figure 13.b). Other possibility would be to burn manually a set of grid nodes linking the spheres. The idea in this case is that the new set of burnt grid nodes generates a connected combinatorial manifold. A similar idea can be implemented for manual split.

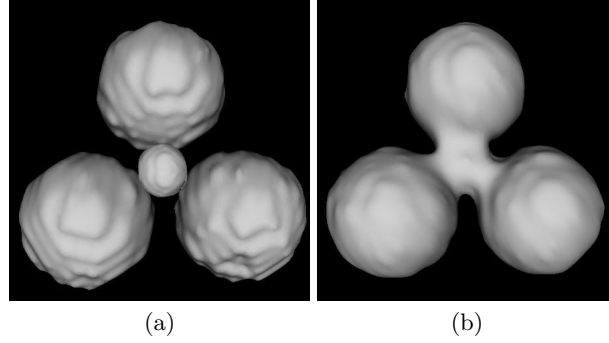


Fig. 13. (a)Original objects. (b)Merge through the user interaction method.

This framework is useful for prosthesis design due to the following aspects. When working in a coarser image resolution we are discarding details in the reconstructed surfaces which may simplifies the drawing of the boundary links in Figure 3. On the other hand, once resolved the segmentation in a coarser resolution we can update the result for a finer image resolution and to get a segmentation with more details. So, we do not lose precision in the whole process. Besides, once completed the geometry reconstruction of the defective skull, we can turn off the T-Surfaces reparameterization (T-Surfaces becomes a mass-spring system). Then, we apply the algorithm summarized in section 2 for obtain a triangulation of the target surface but using a smoothing process based on the mass-spring system instead of the umbrella-operator. Finally, if user interaction is required to improve the prosthesis geometry, we can turn on simplicial domain decomposition framework and allow the user to burn (or unburn) some grid nodes to mimics material addition or removal.

5.1 Multiscale and Subdivision Schemes

The application of the multiscale schemes for digital prosthesis design follows the idea that performing tasks in coarser scales may be easier than in finer ones. For instance, let us return to the Figure 3 and to the problem of finding the internal and external boundary links of the hole. It is not a trivial task specially if the portion of the surface nearby the hole has too many details due to the surface reconstruction process. So, if we apply a multiscale scheme we can reduce details and simplify the definition of these curves.

To be more formal, let us consider parametrically defined curves:

$$\mathbf{x}(t) = (f^1(t), f^2(t), f^3(t)); \quad t \in \mathfrak{R}.$$

Then the wavelet decomposition of the curve is obtained by the decomposition of each coordinate function:

$$f^k(t) = \sum_{i,j} \langle f^k, \psi_{i,j} \rangle \psi_{i,j}(x).$$

The wavelet coefficients of the curve are given by:

$$c_{i,j}^k = \langle f^k, \psi_{i,j} \rangle.$$

We can similarly define the scaling decomposition and coefficients by:

$$f^k(t) = \sum_{i,j} \langle f^k, \varphi_{i,j} \rangle \varphi_{i,j}(x).$$

$$d_{i,j}^k = \langle f^k, \varphi_{i,j} \rangle.$$

From expression (29) it is clear that:

$$P_{m-1}(f^k) = Q_m(f^k) + P_m(f^k). \quad (51)$$

This expression is the starting point for a multiscale approximation of the curve. The scaling coefficients obtained by the projection $P_m(f^k)$ define the multiscale approximation of the curve at level m and the error between the two approximation at scale level m and $m - 1$ is given by the wavelet coefficients computed through the projection $Q_m(f^k)$. A similar process can be defined for parametrically defined surfaces through tensor product of scaling functions [32]. The fact that $P_m(f^k)$ is a approximation of f^k in a coarser scale; that means, with less details, can be explored in order to simplify the search for structures of interest.

Just to clarify the ideas, let us consider the Figure 14 which pictures a (generic) multiscale representation of a function I at scales $\sigma_0, \sigma_1, \dots, \sigma_4$. We observe that the singular points (local maxima and minima) in coarser scales can be easier identified than the corresponding points in the finer scales. So, following expression (51), the key idea when using wavelet representation is to

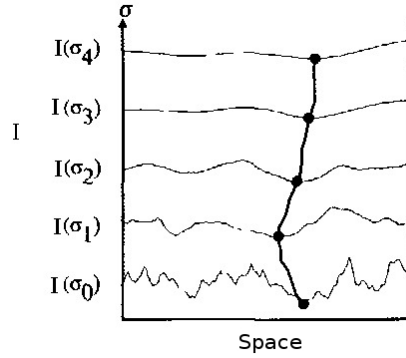


Fig. 14. Scale space representation (Reprinted from [47]).

get the target points in the coarser scale and then to correct its position, using the error $Q_m(f^k)$.

Once the boundaries of the hole are drawn (Figure 3), the subdivision scheme described on section 3.6 can be also applied to generate the two surfaces that, when combined with the hole's surface, complete the prosthesis model. Such process will generate two polygonal surfaces with quadrilateral patches instead of the triangular ones generated by the algorithm described in [49] (section 2). Another aspect is that the user can place guiding points in order to control the geometry of the input net. In the case of the algorithm described in [49], such control is not possible unless we introduce extra machinery.

5.2 The PyImageVis Computer Systems

This section describes the PyImageVis system, an open-source software implemented in Python [6], for processing and visualization of medical images. The main goal of PyImageVis is to provide a computational system for the research of new algorithms for medical image processing and visualization.

In Brazil, the research on visualization and medical image processing is concentrated in universities and research institutes. The INCT-MACC, a virtual institute for applications of scientific computing in medicine, is an example of this fact [2]. The INCT-MACC team need a computational platform which facilitates the validation of new algorithms in image processing and visualization. Softwares such as *Matlab* and *Octave* can be used for this purpose. However, they are limited for visualization of three-dimensional images. In order to meet this requirement, we are developing the *PyImageVis* platform. This software has been implemented in Python language and incorporates functions and methods of existing open source Python libraries, like *NumPy* and *Scipy* [4, 7]. It provides friendly graphical interfaces, which follows the MatLab layout (Figure 15) and is user extendable.

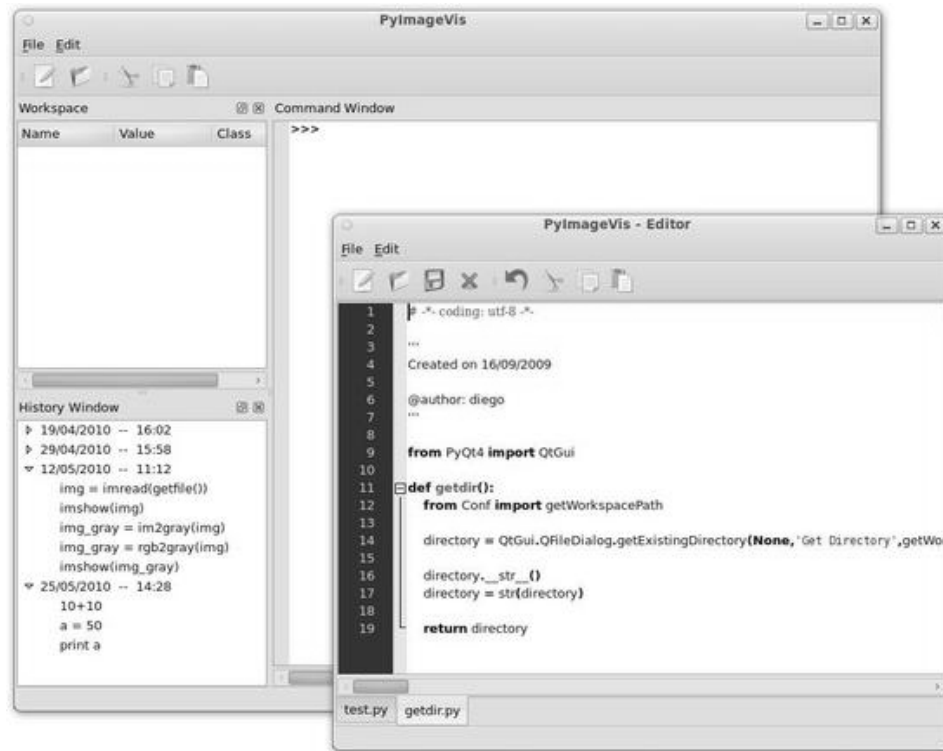


Fig. 15. Graphical user interface components of *PyImageVis*: *Command Window*, *History Window*, *Workspace* and *Python Editor*.

We are incorporating computational resources for digital prosthesis design in the *PyImageVis* system. Figure 16 pictures the main software interface and an application example for isosurface visualization in the case of a defective skull. The image volume is composed by 108 slices, in the *DICOM* format, 512×512 pixel resolution [5].

Figure 17 describes the capabilities of the *PyImageVis* to reconstruct the skull slice-by-slice in the image space. The Figure 17.b shows the result obtained by reflecting the hole image respect to axis of symmetry of the image pictured in Figure 17.a. Besides, we also demonstrate the *PyImageVis* potential with the execution of another slice-by-slice processing task conducted by user interaction. In this case, the user firstly places a paths linking terminal points of the lesion in Figure 18.a, obtaining the boundary of a ribbon like the one in the Figure 18.b. Then, the system applies a region filling to generate the ribbon that completes the lost part of the bone (Figure 18.c).

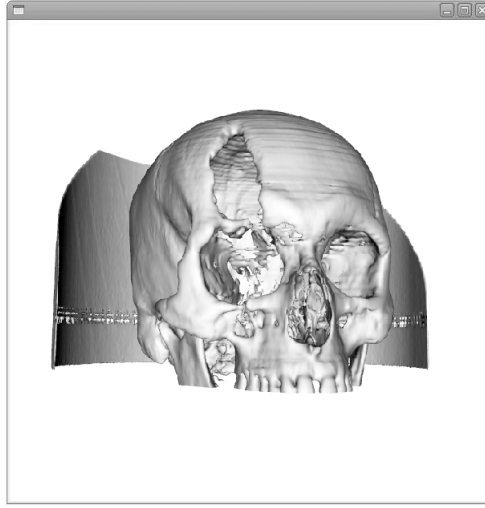


Fig. 16. Surface reconstruction through marching cubes in the *PyImageVis* software.

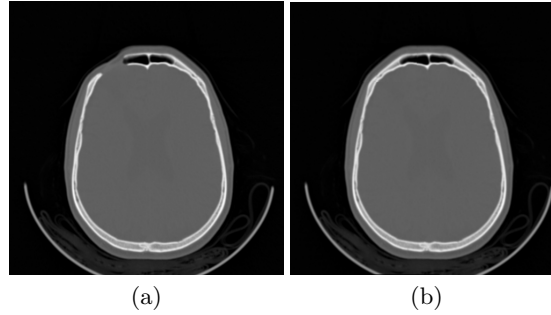


Fig. 17. (a) Original slice of defective skull. (b) Result after reflection.

6 Discussion and Perspectives

An question in this area is how to incorporate prior knowledge about the skull shape in each slice. This question needs some kind of learning process to be addressed. The theory presented on section 3.2, based on PCA, is a possibility in this direction.

First, 40 slices of a normal cranium are used to compose a data set. Figure 19 shows the first, middle and last (segmented) slices. Then, we compute the skeleton for each slice and fit a cubic spline for each obtained skeleton. Then, we could calculate a set of 100 uniformly spaced landmarks for each curve composing a database of 40 curves. Then, we align each curve with the middle one (curve generated from Figure 19.b) in order to discard variations due to rigid transformations (rotation, scale and translation).

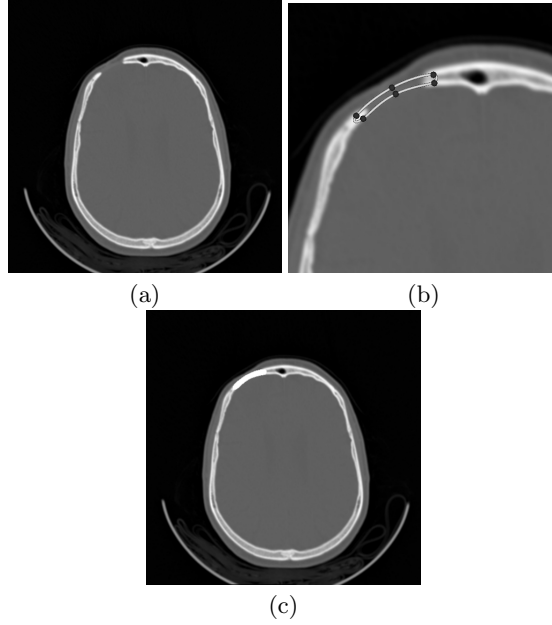


Fig. 18. (a) Original slice with lost part. (b) User defined ribbon. (c) Repaired figure.

The aligned curves are used to compute the mean shape and covariance matrix in expressions (12) and (13), respectively. The spectrum of the covariance matrix is pictured on Figure 20. This figure shows that as the dimension of the PCA most expressive subspace increases, there is an exponential decrease in the amount of total variance explained by the first principal components with the largest eigenvalues.

This is a well-known behavior of the dimensionality reduction provided by the standard PCA [22]. More specific, the first 20 principal components have eigenvalues λ in the range $[2 \leq \lambda \leq 7800]$ and for the first 5 principal components this range is $[184 \leq \lambda \leq 7800]$. The Figure 21 shows the mean curve $\bar{\mathbf{x}}$ and the curve obtained by the expression:

$$\mathbf{x} = \bar{\mathbf{x}} + \alpha \cdot \phi_1, \quad (52)$$

with $\alpha = 50.0$.

We observe that the total variance explained by the PCA most expressive principal component captures the basic shape of the database. This can be further used to implement previous knowledge to the digital prosthesis design process. Fundamentally, we may enforce a limitation that the skeleton of the rebuilt format has to belong to the CA subspace spanned, for example the first 5 main components.

The section 5.1 proposes the application of subdivision methods as an alternative to generate the prosthesis representation. Such process generates surfaces

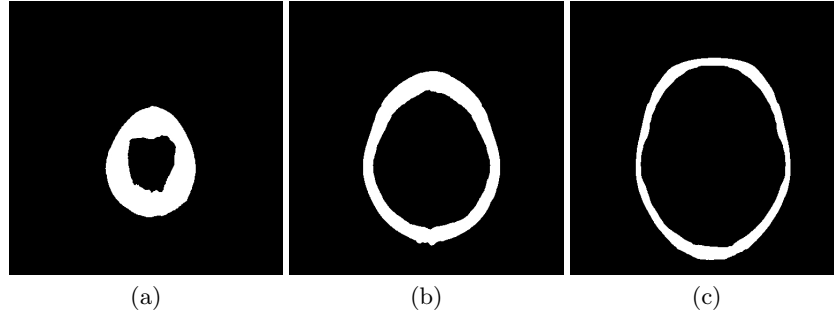


Fig. 19. Bone segmentation results for: (a) First image of the data set. (b) The middle one. (c) The last image.

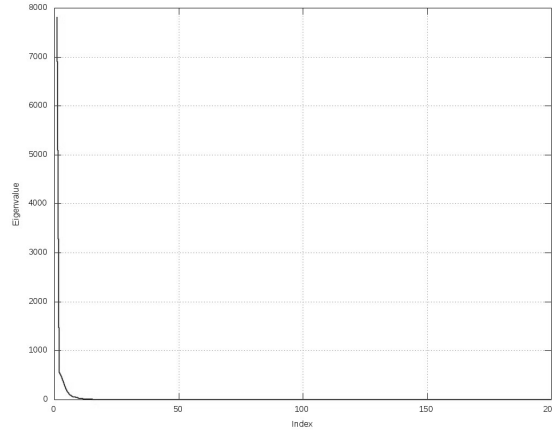


Fig. 20. Exponential decrease of the eigenvalues of the covariance matrix.

with quadrilateral patches instead of the triangular ones generated by the algorithm described in [49]. From Figure 22, We observe that the triangle density of the latter is very height and not uniform over the surface. This feature may be a problem if we need user interaction to fix the curvature in some places. We must consider this intervention because it is very hard to account for all the clinical aspects when customizing a prosthesis. Specifically, we intend to apply the mass-spring system to implement this task. Such variation in the density of triangles implies that the system may be more rigid in some places which is not interesting for user interaction.

The reflection is a very useful method but has some issues also. The Figure 23.a shows the original image and Figure 23.b shows the result of the reflection. We observe some defects that must be fixed. This can be accomplished by user interaction through mouse-guided intervention. Other possibility is to give the

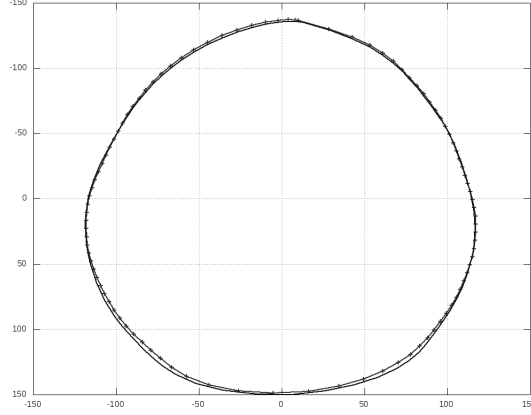


Fig. 21. Mean curve (marked one) and the curve generated by expression (52).

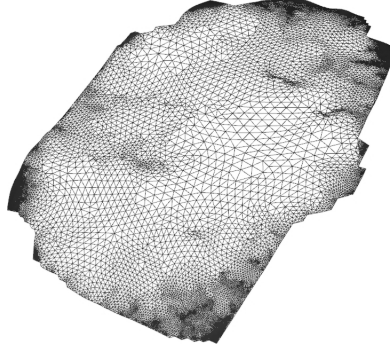


Fig. 22. Triangulated surface generated by the algorithm proposed in [49].

user a tool to construct a ribbon to fix the lesion, like in the Figure 18. However, this process may tedious and imprecise even if we apply the result obtained for a slice, say slice k , and use it to initialize the process in the slice $n + 1$.

On the other hand, the method presented in section balloon depends on the boundary conditions to set up the deformable model. The straightforward way to get this information is through the skeleton of the bone region. Figure 24 shows one examples of skeleton that helps us to understand that this process has also some drawbacks. In this case, if we follow the tangent at the end points of the skeleton we obtain a balloon solution far from the target.

The application of implicit formulations for deformable models is an interesting perspective in this area. As pointed out in [41] we can implement the framework described in section 5 using the level set model [42]. This alternative is interesting not only due to the capabilities of level set but because its imple-

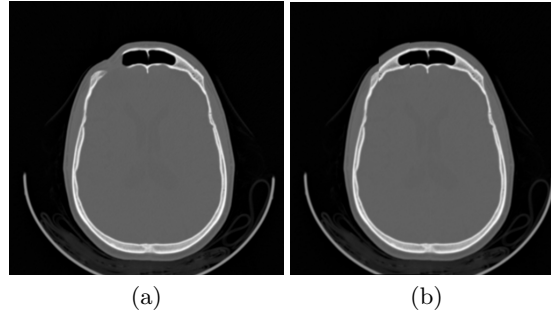


Fig. 23. (a) Original image. (b) Result of reflection.

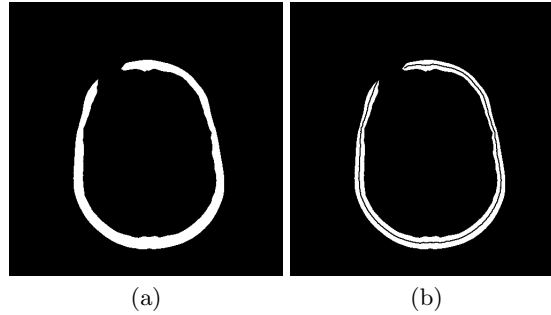


Fig. 24. (a) Segmented bone region. (b) Skeleton with end points unsuitable to get boundary conditions.

mentation is simpler than for T-Surfaces. In fact, the embedding function for the level set plays a similar role of the characteristic function for T-Surfaces (see [39] for details). However, the update of the latter is based on exhaustive tests [31] while the former is just a consequence of the governing equation of the level set method. As a practical consequence, we observed that it is easier to implement the level set than the T-Surfaces which is a motivation for further applications of level set in the prosthesis modeling problem.

7 Conclusions

In this chapter we review approaches to reconstruct the defective region of a skull. The main goal is to automatically construct the prosthesis model for the defective region.

Re-building a new cranium after injury may be achieved through surgical methods by implanting a customized prosthesis. Lately, the creation of digital prosthesis and surgical planning have been done by using image processing, surface rebuilding and geometric techniques. In this chapter we review state-of-the-art approaches in this field and discuss related issues.

The field of prosthesis modeling may include methods for segmentation and surface reconstruction, geometric modeling, multiscale methods and user interaction approaches. We show a promising result based on the balloon model and compare our technique with a state-of-the-art one, showing that our method can generate a more suitable prosthesis geometry. Besides, we discuss perspectives in this area in order to perform all the tasks in 3D space as well as to incorporate prior knowledge in the process.

Acknowledgment

The authors would like to thank CNPq (grant 133865/2009-6) and INCT-MACC for financial support.

References

1. 3-matic for biomedical research. In <http://www.materialise.com/BiomedicalRnD/3-matic>.
2. INCT-MACC - Brazilian Virtual Institute for Applications of Scientific Computing in Medicine. In <http://macc.lncc.br/>.
3. Mimics. In <http://www.materialise.com/mimics>.
4. NumPy. In <http://numpy.scipy.org/> Acessado em: 15 dez. 2009.
5. Portal of brazilian public software. In <http://www.softwarepublico.gov.br>.
6. Python Programming Language. In <http://www.python.org> Acessado em: 18 dez. 2009.
7. Scipy. In <http://www.scipy.org/> Acessado em: 15 dez. 2009.
8. E. L. Allgower and K. Georg. *Numerical Continuation Methods: An Introduction*. Springer-Verlag Berlin Heidelberg, 1990.
9. Gill Barequet and Micha Sharir. Filling gaps in the boundary of a polyhedron. *Computer Aided Geometric Design*, 12(2):207–229, 1995.
10. D. Bhargava, P. Bartlett, J. Russell, M. Liddington, A. Tyagi, and P. Chumas. Construction of titanium cranioplasty plate using craniectomy bone flap as template. *ACTA NEUROCHIRURGICA*, 152(1):173–176, Jan 2010.
11. B. P. Carneiro, C. T. Silva, and A. E. Kaufman. Tetra-cubes: An algorithm to generate 3D isosurfaces based upon tetrahedra. In *International Symposium on Computer Graphics, Image Processing and Vision (SIBGRAPI'96)*, 1996.
12. E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, 1978.
13. Yi-Jen Chiang, C. Silva, and W. Schroeder. Interactive out-of-core isosurface extraction. In *IEEE Visualization 1998*, pages 167–174, 1998.
14. L. D. Cohen. On active contour models and balloons. *CVGIP:Image Understanding*, 53(2):211–218, March 1991.
15. N. Di Lorenzo and V. Pirillo. From computerized tomography data processing to rapid manufacturing of custom-made prostheses for cranioplasty case report comment. *Journal of Neurosurgical Sciences*, 52(4):116, DEC 2008.
16. G. Giraldi, E. Strauss, A. Apolinario, and A. F. Oliveira. An initialization method for deformable models. In *5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001)*, 2001.

17. Gilson A. Giraldi, Antonio A. F. Oliveira, and Leonardo Carvalho. Mining cellular automata databases through pca models. <http://arxiv.org/abs/cs/0511052>, 2005.
18. Marcelo Coelho Goiato, Rodolfo Bruniera Anchieta, Murillo Sucena Pita, and Daniela Micheline dos Santos. Reconstruction of skull defects: Currently available materials. *Journal of Craniofacial Surgery*, 20(5):1512–1518, Set 2009.
19. A. Graps. Historical perspective. http://www.amara.com/IEEEwave/IW_history.html, 2004.
20. D. H. House and D. E. Breen, editors. *Cloth Modeling and Animation*. A K Peters, Ltd., 2000.
21. Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., 1989.
22. R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis*. New Jersey: Prentice Hall, 1998.
23. Chua Chee Kai, Chou Siaw Meng, Lin Sing Ching, Eu Kee Hoe, and Lew Kok Fah. Cranioplasty using polymethyl methacrylate prostheses. *Journal of Clinical Neuroscience*, 16(1):56–63, 2009.
24. Ra Lauric and Sarah Frisken. Soft segmentation of ct brain data. Technical Report TR-2007-3, Department of Computer Science – Tufts University, Medford, MA, 2007.
25. Ming-Yih Lee, Chong-Ching Chang, Chao-Chun Lin, Lun-Jou Lo, and Yu-Ray Chen. Custom implant design for patients with cranial defects. *Engineering in Medicine and Biology Magazine, IEEE*, 21(2):38–44, Março 2002.
26. Sai-Cheung Lee, Chieh-Tsai Wu, Shih-Tseng Lee, and Po-Jen Chen. Rapid prototyping assisted surgery planning. *The International Journal of Advanced Manufacturing Technology*, 14(9):624–630, 1998.
27. A. Levin. Filling n-sided holes using combined subdivision schemes. In Pierre-Jean Laurent, Paul Sablonniere, and Larry L. Schumaker, editors, *Curve And Surface Design*, pages 221–228. Vanderbilt University Press, Nashville, TN, 1999.
28. Liulan Lin, Jiafeng Zhang, and Minglun Fang. Modelling the bio-scaffold for repairing symmetrical and unsymmetrical defective skull. In *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on*, pages 905–908, 16-18 2008.
29. Lin Liulan, Liu Hanqing, Hu Qingxi, Li Limin, and Fang Minglun. Research of the method of reconstructing the repair bionic scaffold based on tissue engineering. *IET Conference Publications*, 2006(CP524):1275–1279, 2006.
30. W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.
31. T. McInerney and D. Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Trans. on Medical Imaging*, 18(10):840–850, October 1999.
32. Y. Meyer. *Wavelets and Operators*. Cambridge University Press, 1992.
33. P. Laeven J. Poukens M.M.A. Beerens. 3-matic assists in true rapid design and manufacturing of a custom skull implant. Technical report, Materialize, www.materialise.com/download/en/2706472/file, 2003.
34. Ron Pfeifle and Hans-Peter Seidel. Triangular b-splines for blending & filling of polygonal holes. In *Graphics Interface*, pages 186–193, Maio 1996.
35. Oleg S. Pianykh. *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*. Springer-Verlag Berlin Heidelberg, 2008.
36. Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*. Springer, 2000.

37. M. Sabin. Cubic recursive division with bounded curvature. In P. J. Laurent, A. le Mehaute, and L. L. Schumaker, editors, *Curves and Surfaces*, pages 411–414. Academic Press, 1991.
38. Abhay MD Sanan and Stephen J. MD Haines. Repairing holes in the head: A history of cranioplasty. *Neurosurgery*, 40(3):588–603, March 1997.
39. J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press, 1996.
40. E. Strauss, W. Jimenez, G. Giraldi, R. Silva, and A. Oliveira. A surface extraction approach based on multi-resolution methods and T-surfaces framework. Technical report, National Laboratory for Scientific Computing, <ftp://ftp.lncc.br/pub/report/rep02/rep1002.ps.Z>, 2002.
41. E. Strauss, W. Jimenez, G. A. Giraldi, R. Silva, and A. F. Oliveira. A semi-automatic surface reconstruction framework based on T-surfaces and isosurface extraction methods. In *International Symposium on Computer Graphics, Image Processing and Vision (SIBGRAPI'2002)*, 2002.
42. Jasjit S. Suri, Kecheng Liu, Sameer Singh, Swamy Laxminarayan, Xiaolan Zeng, and Laura Reden. Shape recovery algorithms using level sets in 2-d/3-d medical imagery: a state-of-the-art review. *IEEE Transactions on Information Technology in Biomedicine*, 6(1):8–28, 2002.
43. P. Sutton and C. Hansen. Accelerated isosurface extraction in time-varying fields. *IEEE Trans. Vis. and Comp. Graphics.*, 6(2):98–107, April-June 2000.
44. B. van Ginneken, A.F. Frangi, J.J. Staal, B.M. ter Haar Romeny, and M.A. Viergever. Active shape model segmentation with optimal features. *IEEE Transactions on Medical Imaging*, 21(8):924–933, 2002.
45. P. Volino and N. Magnenat-Thalmann. *Virtual Clothing: Theory and Practice*. Springer, 2000.
46. Y. Wang. *The Study of Three-dimensional Reconstructed Bioceramics in the Successional Defect of Mandible*, Shanghai: The Second Military Medical University. 2005.
47. G. Witten. Scale space tracking and deformable sheet models for computational vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(7):697–706, July 1993.
48. Y. Zhao and J. Zhou. The analysis of clinical effect of individualized cranioplasty with different materials for 75 skull defect patients. *Medical Journal of Chinese People's Liberation Army*, 31:354–355, April 2006.
49. Fei You, Qingxi Hu, Yuan Yao, and Qi Lu. A new modeling method on skull defect repair. *Measuring Technology and Mechatronics Automation, 2009. ICMTMA '09. International Conference on*, 1(11-12):568–572, April 2009.



Gilson A. Giraldi received his PhD degree in Computer Graphics from Federal University of Rio de Janeiro, Brazil, in 2000. Since then he has been with the National Laboratory for Scientific Computing, Brazil, where he is responsible for academic research projects in the areas of image segmentation and data visualization.



Luiz C. M. de Aquino received his Diploma in Mathematics from Feira de Santana State University in 2004 and he is taking his M.Sc. degree from National Laboratory for Scientific Computing.



Paulo S. S. Rodrigues received his PhD degree in Computer Science from Federal University of Minas Gerais 2003. From 2003 to 2006 he worked in National Laboratory for Scientific Computing. Since 2007 he is Professor of FEI University, São Bernardo do Campo, São Paulo. His main interest areas are pattern recognition and medical image analysis.



Antonio L. Apolinario Jr. received the Doctor in Science degree from COPPE/UFRJ. He is currently a Professor in the Technology Department of State University of Feira de Santana, Bahia. His research areas are: Computer Graphics, Visualization, Computer Animation, Augmented Reality, GPU programming and Computer Games.



Jaime S. Cardoso received his Ph.D. in Computer Vision from Universidade do Porto in 2006. Cardoso is currently a Project Leader in the Telecommunications and Multimedia Unit of INESC Porto, where he leads a research team on machine learning, computer vision and digital signal processing. He is also Assistant Professor at the Faculty of Engineering, University of Porto.



Jasjit Suri, PhD, MBA, Fellow AIMBE, is an innovator, scientist and an internationally known world leader in Biomedical Imaging Sciences and Devices – applied to Diagnostics and Therapeutics. He has published over 300 publications, 22 books, 60 innovations (patents) and 4 FDA clearances covering several fields of medicine. He has been Senior Director and Chief Technology Officer of Fischer Imaging, Eigen Inc and then for Biomedical Technologies. His main interests are in the area of biomedical engineering and its management.