Representation, Shape, Topology and Evolution of Deformable Surfaces. Application to 3D Medical Image Segmentation

Johan Montagnat, Hervé Delingette, Nicolas Scapel, Nicholas Ayache

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Representation, shape, topology and evolution of deformable surfaces. Application to 3D medical image segmentation.

Johan Montagnat — Hervé Delingette — Nicolas Scapel — Nicholas Ayache

N° 3954

Mai 2000

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--3954--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Representation, shape, topology and evolution of deformable surfaces. Application to 3D medical image segmentation.

Johan Montagnat, Hervé Delingette , Nicolas Scapel , Nicholas Ayache

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Epidaure

Rapport de recherche n° 3954 — Mai 2000 — 69 pages

Abstract: These last years, deformable models raised much interest and found various applications in the field of computer vision. They provide an extensible framework to reconstruct shapes. Deformable surfaces, in particular, are used to represent 3D objects. They have been used for pattern recognition [47, 2], computer animation [118], geometric modelling [40, 75], simulation [45], boundaries tracking [14], segmentation [83], etc. In this report we propose a deformable surfaces survey. Many surface representation have been proposed to meet different 3D reconstruction problem requirements. We try to classify the main representations proposed in the literature and we study the effect of the representation on the model evolution behavior, revealing some similarities between different approaches. Whe then focus on a powerful discrete mesh representation, the *simplex meshes*. We propose different algorithms to control simplex meshes shape and topology. Whe show results on 3D medical images segmentation.

Key-words: Deformable Surface, representation, geometry, topology, medical images, 3D

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 —Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Représentation, forme, topologie et évolution des surfaces déformables. Application à la segmentation d'images médicales 3D.

Résumé : Ces dernières années, les modèles déformables ont trouvé de nombreuses applications dans le domaine de la vision par ordinateur. Ils constituent une représentation extensible utile à la reconstruction de formes. Ils sont utilisés dans des applications aussi diverses que la reconnaissance de motifs [47, 2], l'animation [118], la modélisation géométrique [40, 75], la simulation [45], le suivi de contours [14], la segmentation [83], etc. Dans ce rapport, nous détaillons les différentes approches fondées sur les surfaces déformables. De nombreuses représentation ont été proposées pour résoudre des problèmes de reconstruction 3D variés. Nous ordonnons les principales représentations trouvées dans la litérature et nous étudions l'impact de la représentation sur l'évolution des modèles, montrant les similarités existantes entre différentes approches. Nous nous concentrons ensuite sur une représentation souple et élégante: les *maillages simplexes*. Différents algorithmes de contrôle de la forme et de la topologie sont décrits. Quelques résultats des segmentation d'images médicales 3D illustrent la mise en œuvre de ces surfaces déformables.

Mots-clés : Surface déformable, représentation, géométrie, topologie, images médicales, 3D

1 Introduction

For the past decade, there has been a significant research effort for achieving 3D objects modelling based on deformable models. The main motivation of these research works is to provide reliable reconstruction tools that are both robust and generic.

Deformable models cover a very wide range of application and have been used in pattern recognition [47, 2], computer animation [118], geometric modelling [75], surgery simulation [45], tracking [14], image segmentation [86, 61, 5, 59], etc. Extremely variable representations have been used to fulfil different 3D modelling needs, from deformable 3D lines [108, 54] to deformable volumes [121, 22]. In this paper we focus on deformable surfaces that have been vigorously investigated for 3D objects modelling. Deformable models were introduced by Kass *et al* in 2D as explicit deformable contours [60] and generalized to the 3D case by Terzopoulos *et al* [120]. Parametric representations such as superquadrics [117, 10] and discrete representations [89, 42] have also been proposed. Recently, implicit representations have been used with the ability to handle topology changes [74, 130, 69].

When considering the segmentation of images, the basic idea is to steer deformable models towards the image structure boundaries. Due to the image noise and the lack of contrast between structures in many image acquisition systems [1], it is in general needed to constrain the model variation space and to introduce some prior information in the deformation process. For instance, statistical shape variations from a training set [33] may be used to constrain the deformation of a geometric model.

In this report, we propose a survey on existing deformable surfaces. Different approaches are classified according to their surface representations in section 2. Different evolution laws found in the literature are studied in section 3. Although the evolution may depend on the kind of representation used, we exhibit some similarities between different approaches. Surface representation has an effect on the deformation capability of a model. Similarly, modifications of the evolution laws have been introduced to regularize surface deformations. These methods are described in section 4. The ability of various representations to handle topology changes is the topic of section 5. We then focus on a discrete surface representation called *simplex meshes*. It provides a simple geometric representation combined with a powerful framework for regularization (section 6). In section 6.2, several methods to achieve shape regularization are shown. This is a key issue of deformable modelling. Local regularizing constraints are designed to ensure local surface regularity whether prior information on the expected anatomical structures shape is available or not. The deformations are also constrained at a global scale to make the segmentation process robust to outliers and false positives. In section 6.3, an automatic model initialization method is described permitting to generate a surface mesh from an image with the desired shape and topology. The use of topology operators on discrete meshes to allow surface resampling as well as changes in the surface genus is the topic of section 6.4. Finally, the ability of simplex meshes to handle medical image segmentation is shown in section 7. Various image geometries as well as imaging modalities are demonstrated. Heart left ventricle segmentation from 3D ultrasound images are shown.

The wide application field of deformable models has already led to the production of several surveys. McInerney and Terzopoulos [78] study deformable models for medical image analysis and classify the approach found in the literature by applications. Cheung *et al* [21] are interested in deformable models for pattern recognition. Robert [98] details parametric deformable models while Lachaud [63] reviews some deformable models considering simultaneously geometric representation and evolution laws. Gibson and Mirtich [52] propose a general survey of deformable models in computer graphics classified by the method used for computing deformations. Brown [17] focuses on registration methods as well as Audette *et al* [4] who thoroughly classify registration techniques from an algorithmic point of view. This paper focuses on the mathematical and computational foundations of deformable surfaces used for 3D object reconstruction in a wide variety of applications.

2 Deformable surfaces geometry

A deformable surface is an object composed of a surface representation and an evolution law allowing the deformation of this surface to represent a different shape. We first focus on surface geometric representations.

2.1 Shape and Deformation parameters

A key aspect of deformable models lies in their geometric nature. Basically, two criteria may be associated with the geometry of these models :

- **Shape Description.** A deformable model may be restricted to represent simple shapes (for instance ellipsoids) or shapes of restricted topology (for instance spherical Fourier descriptors) or may represent any shapes independently of their topology (for instance spring-mass models).
- **Deformation Description.** The complexity of a model deformation may not be related to the complexity of its shape. For instance, when deforming a surface mesh by applying an affine transformation, the model deformation has only 12 degrees of freedom whereas the mesh shape may be much more complex.

In terms of geometric description, a deformable surface $\mathcal{S}_{\mathbf{q}}$ is controlled by a vector of shape parameters $\mathbf{q} = (q_1, \ldots, q_{n_q})^T$. If the surface has an explicit representation, it is defined as a function :

$$\begin{array}{rccc} \mathcal{S}_{\mathbf{q}} & : & \mathbb{R}^{n_q} \times \Omega & \rightarrow & \mathbb{R}^3 \\ & & (q_1, \ldots, q_{n_q}, r, s) & \mapsto & \mathbf{p_q}(r,s) \end{array} \tag{1}$$

However, in some cases, it may be useful to introduce an additional vector of deformation parameters $\mathbf{d} = (d_1, \ldots, d_{n_d})^T$ that controls the application of a global transformation $T_{\mathbf{d}}$ on the surface :

$$\begin{array}{rccc} \mathcal{S}_{(\mathbf{q},\mathbf{d})} = T_{\mathbf{d}}(\mathcal{S}_{\mathbf{q}}) & : & \mathbb{R}^{n_q} \times \mathbb{R}^{n_d} \times \Omega & \rightarrow & \mathbb{R}^3 \\ & & (q_1, \ldots, q_{n_q}, d_1, \ldots, d_{n_d}, r, s) & \mapsto & T_{\mathbf{d}}\left(\mathbf{p_q}(r,s)\right) \end{array} \tag{2}$$

Usually $T_{\mathbf{d}}$ is chosen as an endomorphism of $\mathbb{R}^3$. It is important to note that we could also include the set $\mathbf{d}$ into the set $\mathbf{q}$ of shape parameters. However, the mathematical methods associated with these two formulations are slightly different (see sections 3.1 and 4 for the two approaches).

Therefore, we can distinguish between two extreme cases. On one hand, when $n_d = 0$ and $n_q \neq 0$, we have a typical *deformable model framework* with a large variety of possible surface representations. These surface representations are described in the rest of section 2. and are illustrated in Figure 1.

On the other hand, when $n_q = 0$ and $n_d \neq 0$, we have a *surface registration framework*, where a surface is deformed through the application of a global transformation :

$$\begin{array}{rcccl} \mathcal{S}_{\mathbf{d}} & : & \mathbb{R}^{n_d} \times \Omega & \rightarrow & \mathbb{R}^3 \\ & & (d_1, \ldots, d_{n_d}, r, s) & \mapsto & T_{\mathbf{d}}\left(\mathbf{p}(r,s)\right) \end{array} \tag{3}$$

In this case, the initial surface description $\mathcal{S}$ does not even have to be a parametric surface. Indeed, this approach may be applied to any surface having an explicit representation even a discrete surface mesh.

In section 4, we describe several approaches combining a global transformation $n_d \neq 0$ and shape parameters $\mathbf{n}_q \neq 0$.

2.2 Continuous and discrete representations

First, we make a distinction between deformable surfaces having continuous and discrete representations. With discrete representations, the geometry of surfaces is only known at a finite set of points. Continuous representations must be discretized for computational needs but they offer the ability to compute differential quantities such as surface normals or curvatures almost everywhere on the surface. In fact, we have extended the notion of discrete representation to include any surface representation having at most a C^0 continuity (position continuity) along the surface. This is why we have included triangulations and simplex meshes in the set of discrete surfaces.

Each family in turn can be divided into different classes of representations. Continuous surfaces may be defined through an explicit or an implicit equation while discrete surfaces are mainly represented through discrete meshes or particle systems.

2.3 Explicit representations

When using an explicit representation, surfaces are described by coordinates functions depending on a vector of shape parameters $\mathbf{q}$ as described in equation 1. A further distinction between these models can be made depending on the number and the nature the parameter set $\mathbf{q}$. The number of parameters has an impact on the complexity of the deformation. Another criterion is the local or global deformation impact of these parameters. With parameters controlling local deformations, fairly complex shapes may be obtained but at the

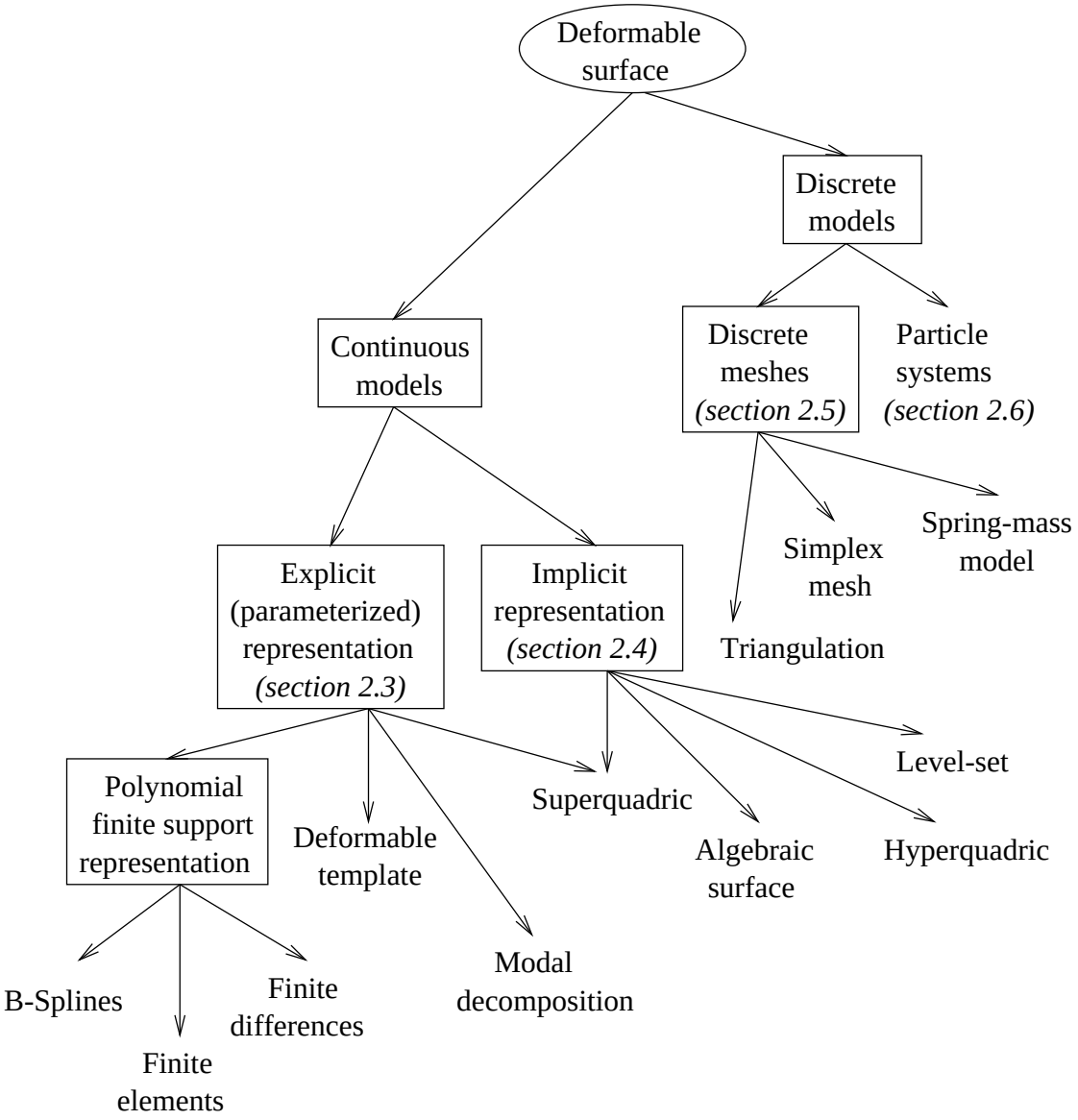

Figure 1: Different geometric representations of deformable surfaces.

expense of an important computational complexity. On the contrary, global shape parameters lead to more stable numerical schemes when a limited number of parameters are estimated.

Polynomial finite support functions

In these representations, shape parameters correspond to the coefficients of polynomial finite support functions. Each parameter has only a local impact on the surface shape. This often leads to solve linear systems of equations with sparse matrices. This category includes the different types of contour and surface splines among which B-splines are the most widely used. In [80], Menet *et al* introduce «*B-snakes*», some deformable contours represented as B-spline curves. B-splines are smooth curves or surfaces with high level of geometric continuity but they allow sharp corners by duplicating some control points to make the curve $\mathcal{C}^0$ only. Menet *et al* use B-snakes for recovering building contours from aerial images. B-spline surfaces have also encountered some success in the deformable models community for registration of anatomical structures [54, 13] and contour tracking [11] in image sequences. Leitner *et al* [67, 66] define deformable contours and surfaces based on B-snakes with the ability to operate topology changes.

In fact most finite element models are based on shape functions that are polynomial finite support functions. These shape functions are usually related to Hermite polynomials in order to guarantee at least a C^1 continuity along the surface. Also, these approaches are often limited to represent surfaces of simple topologies because of their difficulty to ensure C^1 or G^1 geometric continuity on surfaces of arbitrary topology (see for instance [48]) while keeping local support functions.

Finite elements methods are mainly used as a numerical framework for minimizing deformation energy functionals. They are further studied in section 3.4.

Superquadrics

Superquadrics are a generalization of quadrics to surfaces. They certainly are the most widely used parameterized surfaces [117, 10, 97, 124]. In particular, superellipsoids represent closed surfaces. A superellipsoid is defined by the parametric equation :

$$\mathcal{Q}_{\mathbf{q}}(r,s) = \begin{cases} a_1 \text{sign}(\cos r \cos s) |\cos r \cos s|^{\epsilon_1} \\ a_2 \text{sign}(\cos r \sin s) |\cos r \sin s|^{\epsilon_2} \\ a_3 \text{sign}(\sin s) |\sin s|^{\epsilon_3} \end{cases},$$

where $\mathbf{q} = (a_1, a_2, a_3, \epsilon_1, \epsilon_2, \epsilon_3)^T$ is the surface parameter vector, $r \in [0, 2\pi[$ and $s \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Superquadrics allow the representation of relatively variable shapes but only with a symmetry axis. Thus they cannot, in their primitive formulation, represent complex shapes.

Superquadrics can also be represented as implicit surfaces by the equation :

$$F_{\mathbf{q}}(\mathbf{p}) = \left|\frac{p_x}{a_1}\right|^{2/\epsilon_1} + \left|\frac{p_y}{a_2}\right|^{2/\epsilon_2} + \left|\frac{p_z}{a_3}\right|^{2/\epsilon_3} = 1.$$

The surface is defined as $\mathcal{S}_{\mathbf{q}} = \left\{\mathbf{p} \in \mathbb{R}^3 | \mathrm{F}_{\mathbf{q}}(\mathbf{p}) = 1\right\}$. This representation is suited to determine if a point $\mathbf{p} \in \mathbb{R}^3$ belongs to the interior ($F_{\mathbf{q}}(\mathbf{p}) < 1$) or the exterior ($F_{\mathbf{q}}(\mathbf{p}) > 1$) of

the surface. Choosing one representation instead of the other affects the evolution behavior of the model as we can see later in this report.

Deformable templates

In their paper, Yuille *et al* [129] define a parametric model suited to the representation of the eye. The set of shapes that the model can represent is restricted to the specific needs of the application. It overcomes the shape variation restrictions by using only parameters adapted to a given application while making the evolution process robust since the number of parameters is as small as possible.

Modal decomposition

Different representations use a modal decomposition of the model. The decomposition basis is a set of different frequency harmonics. This representation is equivalent to a parametric surface set whose parameters are the mode weights. The sum of the first modes composing the surface gives a rough approximation of its shape. The approximation refines as higher frequency modes are added to the representation. In practice, it is desired to reduce as much as possible the number of modes used to obtain a compact representation, knowing that few modes usually allow the representation of rather complex shapes. Moreover, the small number of modes allows the regularization of the surface when considering its evolution [89, 93].

Staib *et al* [105] investigate Fourier mode decomposition. A curve may be represented as a sum of sinusoidal terms:

$$\mathcal{C}_{\mathbf{q}}(r) = \begin{pmatrix} a_0 \\ c_0 \end{pmatrix} + \sum_{k=1}^{\infty} \begin{pmatrix} a_k & b_k \\ c_k & d_k \end{pmatrix} \cdot \begin{pmatrix} \cos kr \\ \sin kr \end{pmatrix},$$

where the parameter vector is $\mathbf{q} = (a_0, \ldots, a_K, b_1, \ldots, b_K, c_0, \ldots, c_K, d_1, \ldots, d_K)^T$ if only K modes are used. This representation extends to surfaces by using spherical harmonics, for instance, as in Székely *et al* [112]:

$$\mathcal{S}_{\mathbf{q}}(r,s) = \sum_{k=0}^{\infty} \sum_{m=-k}^{k} c_k^m Y_k^m(r,s),$$

where Y_k^m is a spherical harmonic of degree k and order m, and $\mathbf{q} = (c_0^{-n}, \ldots, c_n^n)^T$. Other decomposition bases may be used. To simplify the surface parameterization, Staib and Duncan [105] propose different bases depending on the surface topology (surfaces homeomorphic to a sphere, a torus, a cylinder or a plane).

2.4 Implicit representation

Contrary to explicit models, implicit representations involve an implicit equation to locate the surface points. An implicit surface is generally defined as the zeros' set of a function f valued in $\mathbb{R}$. That is:

$$\mathcal{S}_f = \left\{ \mathbf{p} \in \mathbb{R}^3 | \mathrm{f}(\mathbf{p}) = 0 \right\}. \tag{4}$$

Algebraic surfaces

Algebraic surfaces are surfaces satisfying equation 4 where f is a polynomial. They have been widely used in the field of graphics and modelling [114]. Taubin *et al* [115] use algebraic surfaces to reconstruct unstructured point set data. Algebraic surfaces have several limitations. They are not necessarily closed, although it is possible to consider only some even order polynomials. Algebraic surfaces are not straightforward to display on a graphic screen. Moreover computing the distance from a point to an algebraic surface, which is required by surface matching algorithms, is difficult. The value $f(\mathbf{p})$ is often used as an approximation.

Superquadrics

Superquadrics presented in section 2.3 may be represented in an implicit form. In particular, superellipsoids with implicit equation [9]:

$$F_{\mathbf{q}}(\mathbf{p}) = \left(\left(\left(\frac{p_x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{p_y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{p_z}{a_3} \right)^{\frac{2}{\epsilon_1}} \right)^{\frac{\epsilon_1}{2}} = 1 \tag{5}$$

represent closed surfaces.

Hyperquadrics

Hyperquadrics [55, 25] have been proposed as a superquadric extension. They allow the representation of a wider surface family. An hyperquadric is defined by equation:

$$f_{\mathbf{q}}(\mathbf{p}) = \sum_{j=1}^{n \geq 3} |a_j p_x + b_j p_y + c_j p_z + d_j|^{\epsilon_j} = 1,$$

where $\epsilon_j > 0$ for all j, and $\mathbf{q} = (a_1, b_1, c_1, d_1, \epsilon_1, \ldots, a_n, b_n, c_n, d_n, \epsilon_n)^T$. This equation does not have any explicit representation as soon as $n > 3$. The surface is enclosed in the convex envelope defined by the set of planes $a_j p_x + b_j p_y + c_j p_z + d_j = \pm 1$. As ϵ_j coefficients increase the surface merges with its convex envelope. As well as modal representation, hyperquadrics is an extensive formulation allowing the iterative addition of terms that refine the surface description. However, a hyperquadric must remain homeomorphic to a sphere.

Level sets

Level set methods were developed by Osher and Sethian [91] and introduced in the medicine vision community by Malladi *et al* [74]. Level set methods are thoroughly described in [101]. The main idea of level sets is to embed the deformable model in higher dimension space. A surface is represented as the zero level set of a function $\Psi : \mathbb{R}^3 \rightarrow \mathbb{R}$:

$$\mathcal{S} = \{\mathbf{p} \in \mathbb{R}^3 | \Psi(\mathbf{p}) = 0\}.$$

Thus $\mathcal{S}$ is necessarily a closed surface or a set of closed surfaces.

Given an initial surface $\mathcal{S}_0$, the hypersurface Ψ is defined by :

$$\Psi(\mathbf{x}) = \text{dist}(\mathbf{p}, \mathcal{S}_0),$$

where «dist» is the signed Euclidean distance between a point and the surface. The distance is positive if the point lies outside the surface and negative otherwise. The evolution of surface $\mathcal{S}$ is guided by a partial differential equation involving function Ψ. Its extension to higher dimension space is straightforward.

The main advantage of level set methods is to naturally make surface topology changes. $\mathcal{S}$ may split in several connected components or merge from several components while Ψ remains a function. Their main drawbacks are the computational cost, since a higher dimension space is used for updating the surface, and the lack of user interactivity. Malladi *et al* [73] propose several technics to speed-up the numerical aspects including the use of narrow bands.

Siddiqi *et al* [102] use a propagation equation similar to level sets to recognize the shape of binary structures at different smoothing scales.

2.5 Discrete meshes

The representation of deformable models as discrete entities naturally rises the parameterization problem of continuous representations. However, this approach permits a large number of degrees of freedom to the surface and it is therefore often necessary to constrain the space of possible deformations. Most discrete models are meshes defined as a set of points with some connecting relation.

Discrete contour

A discrete deformable contour $\mathcal{C}$ is just a set of vertices $\{\mathbf{p}_0, \ldots, \mathbf{p}_{d-1}\}$ connected to produce a polygonal line closed or opened. Let $\mathrm{PP}_i : \{0,1\} \rightarrow [0, d-1]$ be the neighborhood relation of vertex i :

$$\begin{cases} \forall i \in [1, d-2], \mathrm{PP}_i(0) = i+1 \text{ and } \mathrm{PP}_i(1) = i-1 \\ \mathrm{PP}_0(0) = 1 \text{ and } \mathrm{PP}_{d-1}(1) = d-2 \\ \mathrm{PP}_0(1) = d-1 \text{ and } \mathrm{PP}_{d-1}(0) = 0 \text{ if } \mathcal{C} \text{ is closed.} \end{cases}$$

Triangulation

A surface is also represented by a set of vertices and a neighborhood relation that induces topological constraints. A common representation are triangulations (see figure 2 on the left) for which the surface is composed of a set of adjacent triangles. Each triangle of a triangulation shares at least one of its edges with a neighbor triangle.

Spring-mass model

Vasilescu and Terzopoulos [122] used spring-mass models composed of a set of nodes connected by springs in the context of surface reconstruction. An example of a spring-mass model where n point masses are linked eight by eight [89] is shown in figure 2 on the right. The springs rest length determines the model rest shape although mass-spring models have several equilibrium positions. The springs' stiffness controls the model elastic properties. It is possible to add springs between vertices that are not direct neighbors in order to improve the mesh stability and to provide a better volumetric behavior.

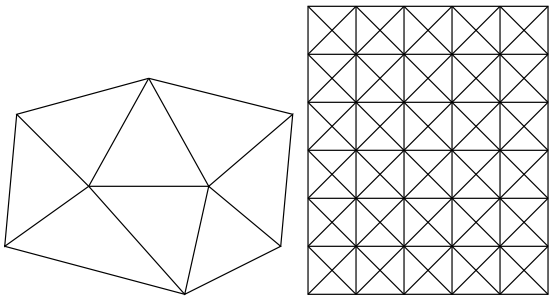

Figure 2: Left: triangulation. Right: mass-spring model.

Simplex meshes

Simplex meshes were introduced by Delingette [39, 42] and studied in [82, 127, 53, 83]. A simplex mesh is a discrete model representation with a constant vertex connectivity. 2-simplex meshes are used to represent surfaces. Each vertex of a 2-simplex meshes is connected to three, and only three, neighbors (see figure 3 on the right). 2-simplex meshes are topologically dual to triangulations as illustrated in figure 3 on the right: there exists a dual triangle for each mesh vertex and a dual triangulation vertex for each mesh face. Due to their constant connectivity, the geometry of simplex meshes is fairly simple. A notion of surface local shape allows the definition of shape memory constraint in the deformation process (see section 6).

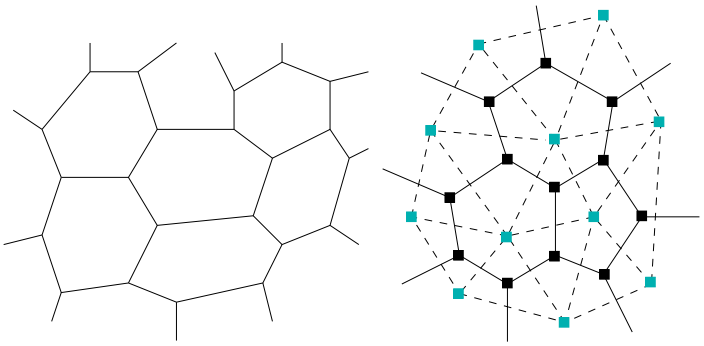

Figure 3: Left: 2-simplex mesh. Right: triangulation dual to a simplex mesh.

Other discrete meshes

Vemuri and Guo [123] use a hybrid model called *pedal snake* composed with a discrete deformable contour and a parameterized generative ellipse. The discrete representation is

also suited for dynamic programming of the model evolution as demonstrated in [3]. Also, subdivision surface techniques [58, 106] allow to describe smooth surfaces by iteratively subdividing an unstructured mesh. However, since the limit surface cannot be recovered explicitly but only approximately as a function of the control points, it often leads to computationally expensive schemes for approximating real data.

2.6 Particle systems

Particle systems are discrete models with a representation fundamentally different to meshes. They are composed of a set of elementary objects called particles. Particles are described by their location, speed, acceleration, mass, and any other parameter needed for a given application. Each particle evolves according to Newtonian mechanical laws. Particles interfere with each other through attraction-repulsion forces that tend to arrange them on a regular lattice (hexagonal grid) when external forces vanish. Figure 4 shows an example of the force amplitude used for a particle system in [68]. Particle systems are suited to represent viscoelastic solids or fluids [46].

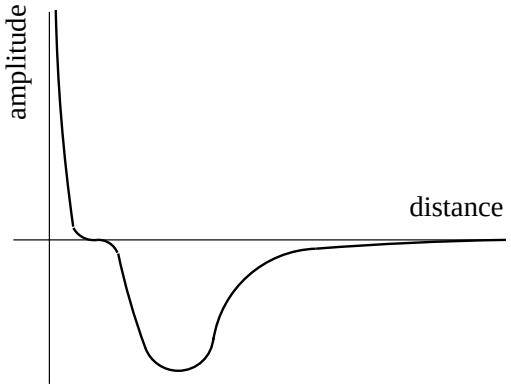

Figure 4: Particle system force amplitude.

In [111], Szeliski and Tonnensen propose oriented particles to represent surfaces. Each particle is considered as a surface element, or *surfel* by analogy to pixels. To each particle is associated a rotation matrix. The authors define coplanar and cocircular energies that tend to align the particles on the surface of a plane or a sphere at rest respectively. Lombardo [68] extends this framework to define a curvature on the surface. He proposes a hybrid model combining a particle system and an implicit surface describing the model envelope.

Particle systems are adapted to represent surfaces with a variable topology. Szeliski and Tonnensen [111] propose ad-hoc rules to dynamically add or remove particles to a system. New particles fill in the surface holes. However, particle systems do not allow a direct computation of the surface geometric properties and are rather complex to visualize.

3 Deformable models evolution

Regardless of the surface representation used, it is needed to deform the model to fit the data. The surface representation affects the possible variability of a model. With few shape parameters, a surface model has restricted shape variations. Implicit representation provides much more freedom to the model that can undergo topology changes but it also makes the model more sensitive to data noise. The deformation process also permits the introduction of restrictions in the allowed shape deformations of a surface.

The deformation process generally involves a data term attracting the model towards structures to reconstruct and a regularization term enforcing a smooth behavior of the model. Data is used to drive the model deformation but the data noise or the outliers make it necessary to constrain the possible deformation space in order to stabilize the model evolution.

A classical approach consists in defining an *energy* functional E. The energy allows the quantification of the model geometric quality (such as its smoothness or its distance to a prior shape) and the data reconstruction accuracy. A regular model fitting the data has a low energy while the energy increases with the model rough shape and its distance to the data. An optimal model thus minimizes E. This minimization requires to differentiate and to discretize the energy equation.

A second approach consists in using directly the derived equation. By analogy with mechanical systems, the derived energy is a force. An ad-hoc set of forces is defined that drives the surface : we then call these forces *differential stabilizers* [37].

Figure 5 sums up the resolution methods described in this section.

3.1 Energy

The energy E of a deformable model is composed of several terms including at least an intrinsic regularizing term E_{reg} and a data term E_{ext}. In the rest of the paper we consider $L^2(\Omega)$, the set of square integrable functions on Ω: $L^2(\Omega) = \left\{ f \in (\Omega \to \mathbb{R}) | \int_\Omega \mathrm{f}^2 < +\infty \right\}$. $L^2(\Omega)$ is characterized by the scalar product $(f, g) = \int_\Omega fg$ and the induced norm $\|f\|^2 = (f, f)$.

The energy of a n-dimensional parameterized surface $\mathcal{S} \in L^2(\Omega)^n$ is defined by:

$$\begin{array}{rccl} E & : & L^2(\Omega)^n & \to & \mathbb{R} \\ & & \mathcal{S} & \mapsto & E(\mathcal{S}) = E_{\text{reg}}(\mathcal{S}) + E_{\text{ext}}(\mathcal{S}) \end{array}$$

We admit that the energy functional is built such that its global minimum coincide with the expected solution $\mathcal{S}_0$:

$$\mathcal{S}_0 = \min_{\mathcal{S}} E(\mathcal{S})$$

The regularizing term has a stabilizer role since the data term is usually very irregular and shows a large amount of local minima. Tikhonov stabilizers are often used to measure

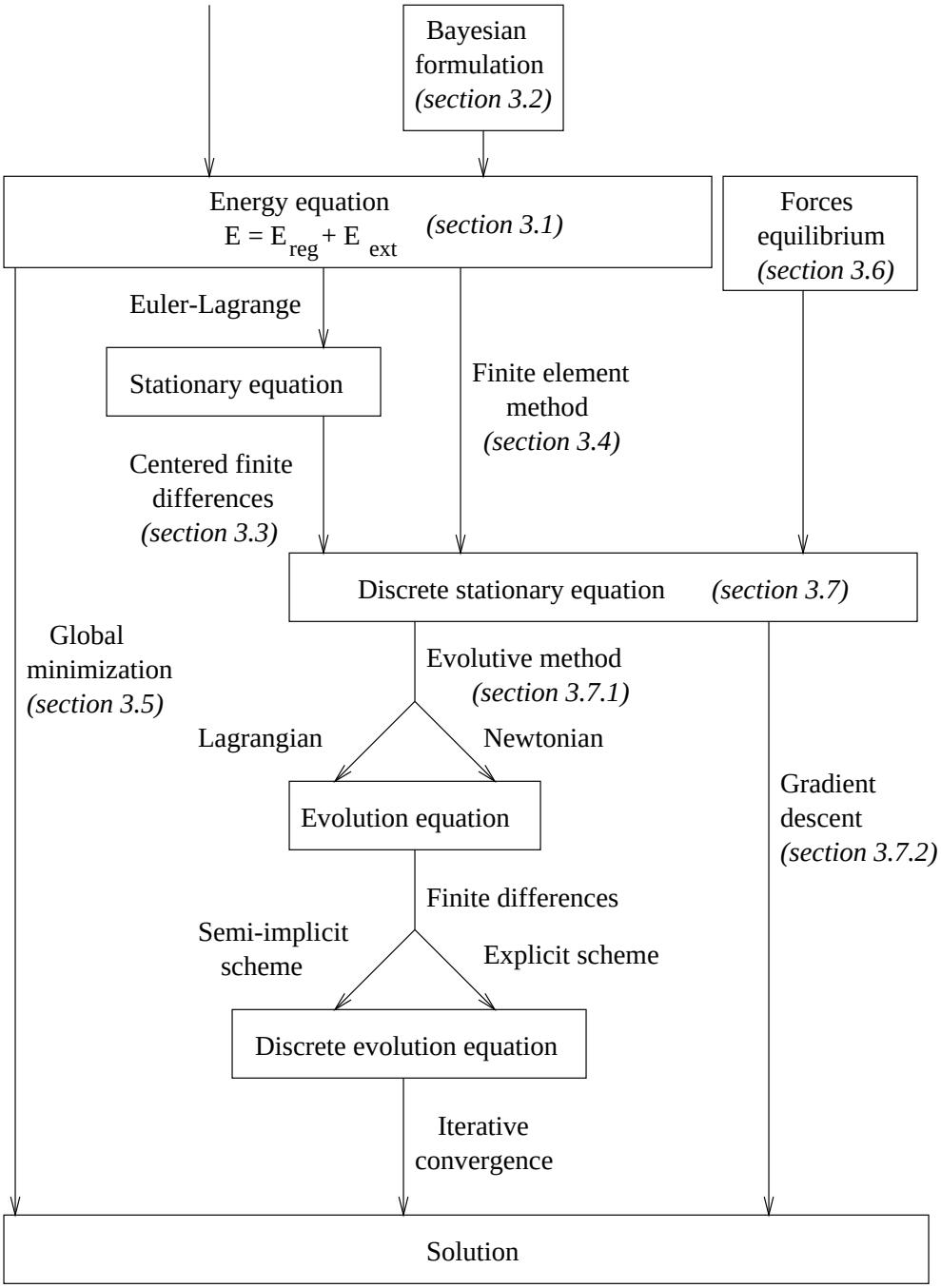

Figure 5: Solving deformable models' evolution equation.

the surface smoothness. Thus, a surface internal energy is:

$$E_{\text{reg}}(\mathcal{S}) = \int_\Omega \sum_{1 \leq i+j \leq k} \frac{(i+j)!}{i!j!} w_{ij}(r,s) \left\| \frac{\partial^{i+j}\mathcal{S}}{\partial r^i \partial s^j} \right\|^2 dr ds \tag{6}$$

where w_{ij} are positive functions. Second order stabilizers are mostly used ($k = 2$). The stabilized surface must thus be twice differentiable, that is $\mathcal{S} \in H^2(\Omega)^3$ where $H^2(\Omega)$ is the Sobolev space defined by:

$$H^2(\Omega) = \left\{ f \in L^2(\Omega) | \forall i,j \ \frac{\partial^2 f}{\partial x_i \partial x_j} \in L^2(\Omega) \right\}$$

A surface regularized by a second order Tikhonov stabilizer is identical to a thin membrane with mechanical energy E_{reg}. Functions w_{10} and w_{01} measure the membrane stress while w_{20}, w_{02} and w_{11} measure its rigidity.

The data term depends on a potential P computed from the data (*e.g.* for an image I, $P = -\|\nabla I\|^2$ in its simplest form):

$$E_{\text{ext}}(\mathcal{S}) = \int_\Omega P(\mathcal{S}(r,s)) dr ds$$

The problem is to determine the surface $\mathcal{S}_0$ such as:

$$\mathcal{S}_0 = \min_{\mathcal{S}} \int_\Omega w_{10} \left| \frac{\partial \mathcal{S}}{\partial r} \right|^2 + w_{01} \left| \frac{\partial \mathcal{S}}{\partial s} \right|^2 + 2w_{11} \left| \frac{\partial^2 \mathcal{S}}{\partial r \partial s} \right|^2 + w_{20} \left| \frac{\partial^2 \mathcal{S}}{\partial r^2} \right|^2 + w_{02} \left| \frac{\partial^2 \mathcal{S}}{\partial s^2} \right|^2 dr ds + \int_\Omega P(\mathcal{S}) dr ds \tag{7}$$

Equation 7 does not admit any analytical solution. It is needed to discretize this equation and to apply an iterative scheme to solve it. The most common method is to rewrite this equation as a stationary equation corresponding to a force equilibrium. The equilibrium equation is then discretized using finite differences. Other resolution methods exist such as dynamic programming [3], greedy algorithms [128] or simulated annealing [94, 107].

3.2 Bayesian approach

Another classical approach to optimize the shape of a surface $\mathcal{S}$ embedded in an image I is to use a statistical framework [129, 109]. Probability $P(\mathcal{S}|I)$ denotes the *posterior statistical model*. It estimates how a surface $\mathcal{S}$ fits the data image I. The searched solution is the *posterior maximum* (MAP), that is the surface that maximizes the *posterior* model probability:

$$\text{MAP} = \max_{\mathcal{S}} P(\mathcal{S}|I)$$

The *posterior* model is estimated by using Bayes rule:

$$P(\mathcal{S}|I) = \frac{P(I|\mathcal{S})P(\mathcal{S})}{P(I)} \tag{8}$$

composed of the following terms:

- $P(I) = \sum_{\mathcal{S}} P(I|\mathcal{S})$ is a normalization constant.
- The *prior model* $P(\mathcal{S})$ is a geometric model measuring the surface shape quality without taking data in account.
- The *data model* $P(I|\mathcal{S})$ measures how the given surface generates the image intensity. This is the stochastic process description linking the unknown state $\mathcal{S}$ with the data I.

The prior model must lead to high probabilities for $\mathcal{S}$ configurations that correspond to a weak regularization energy. A Gibbs (or Boltzmann) distribution is often used:

$$P(\mathcal{S}) = \frac{1}{Z_{\text{reg}}} e^{-E_{\text{reg}}(\mathcal{S})},$$

where Z_{reg} is a regularizing constant called the partition function.

Similarly, a Gibbs distribution may be used for the data model, supposing an acquisition model based on linear measures disturbed by a Gaussian noise:

$$P(I|\mathcal{S}) = \frac{1}{Z_{\text{ext}}} e^{-E_{\text{ext}}(\mathcal{S})}.$$

Finally, the *posterior* model is:

$$P(\mathcal{S}|I) = \frac{1}{Z} e^{-E(\mathcal{S})}.$$

The maximum of the conditional probability $P(\mathcal{S}|I)$ corresponds to the energy minimum.

This approach is interesting if the data term is computed taking the acquisition noise into account. Otherwise, this framework is basically equivalent to the regularization framework described in the previous section [109].

Terzopoulos and Szeliski [119] describe a "Kalman snake" based on a probabilistic modelling by adding a Kalman filter to prior models and data with a Bayesian formulation.

3.3 Finite differences discretization

The energy is usually a non convex function that owns several local minima. The goal is to make the model evolve towards one of those minima. The minimization result thus depends on the initial pose and shape of the model.

Stationary equation

Using Euler-Lagrange equation ($\nabla E(\mathcal{S}) = 0$), it can be proved [24] that an energy local minimum must satisfy:

$$\begin{cases} -\frac{\partial}{\partial r}\left(w_{10}\frac{\partial \mathcal{S}}{\partial r}\right) - \frac{\partial}{\partial s}\left(w_{01}\frac{\partial \mathcal{S}}{\partial s}\right) + 2\frac{\partial^2}{\partial r \partial s}\left(w_{11}\frac{\partial^2 \mathcal{S}}{\partial r \partial s}\right) + \frac{\partial^2}{\partial r^2}\left(w_{20}\frac{\partial^2 \mathcal{S}}{\partial r^2}\right) + \frac{\partial^2}{\partial s^2}\left(w_{02}\frac{\partial^2 \mathcal{S}}{\partial s^2}\right) = -\nabla P(\mathcal{S}) \\ \text{boundary conditions} \end{cases} \tag{9}$$

This equation might be seen as an equilibrium between internal regularizing forces and external data forces. It requires the computation of the surface fourth order derivatives which might cause numerical stability problems. $\mathcal{S}$ must belong to $H^4(\Omega)^p$ space.

A solution of equation 9 can be computed by discretizing this equation using centered finite differences.

Finite differences

The parameter domain Ω is discretized along a regular grid $\Omega_{\Delta r \Delta s}$ of size $M \times N$. The grid nodes are distant of $\Delta r = \frac{1}{M-1}$ in one direction and $\Delta s = \frac{1}{N-1}$ in the other. The function $f(r,s)$ is represented by the value array $f[m,n] = f(m\Delta r, n\Delta s)$ with $0 \leq m \leq M$ and $0 \leq n \leq N$. The discrete derivation operators in the parameter space are defined by centered differences:

$$\frac{\partial f}{\partial r}(mh_1, nh_2) \simeq D_r f[m,n] = \frac{f[m+1,n] - f[m-1,n]}{2\Delta r}$$

$$\frac{\partial f}{\partial s}(mh_1, nh_2) \simeq D_s f[m,n] = \frac{f[m,n+1] - f[m,n-1]}{2\Delta s}$$

$$\frac{\partial^2 f}{\partial r^2}(mh_1, nh_2) \simeq D_{rr} f[m,n] = \frac{f[m+1,n] - 2f[m,n] + f[m-1,n]}{\Delta r^2}$$

$$\frac{\partial^2 f}{\partial s^2}(mh_1, nh_2) \simeq D_{ss} f[m,n] = \frac{f[m,n+1] - 2f[m,n] + f[m,n-1]}{\Delta s^2}$$

$$\frac{\partial^2 f}{\partial r \partial s}(mh_1, nh_2) \simeq D_{rs} f[m,n] = \frac{f[m+1,n+1] - 2f[m,n] + f[m-1,n-1]}{\Delta r \Delta s}$$

After discretizing equation 6, the internal energy appears on a matrix form $\mathbf{K}(\mathcal{S})\mathcal{S}$ where $\mathbf{K}$ designates a $MN \times MN$ rigidity matrix and the force vector $\mathbf{f}(\mathcal{S})$ represents the data potential. The stationary equation thus composes a non-linear system of dependent equations:

$$\mathbf{K}(\mathcal{S})\mathcal{S} = \mathbf{f}(\mathcal{S}). \tag{10}$$

If w_{ij} functions are fixed, the system becomes linear :

$$\mathbf{K}\mathcal{S} = \mathbf{f}(\mathcal{S}). \tag{11}$$

3.4 Discretization using finite elements

The Sobolev space $H_0^2(\Omega)$ is defined by:

$$H_0^2(\Omega) = \left\{ f \in H^2(\Omega) | f_{|\partial\Omega} = 0 \text{ and } \partial f_{|\partial\Omega} = 0 \text{ and } \partial^2 f_{|\partial\Omega} = 0 \right\}$$

where $\partial\Omega$ is the border of Ω. It can be proved [24] that solving equation 9 is equivalent to find a function $v : t \in [0,T] \mapsto v(t) = (\mathbf{u} \in \Omega \mapsto \mathcal{S}(\mathbf{u},t)) \in H_0^2(\Omega)$ such as:

$$\forall u \in H_0^2(\Omega), a(v(t),u) = L_v(u),$$

where a is a bilinear form defined as:

$$a(u,v) = \int_\Omega w_{10}\frac{\partial u}{\partial r}\frac{\partial v}{\partial r} + w_{01}\frac{\partial u}{\partial s}\frac{\partial v}{\partial s} + w_{20}\frac{\partial^2 u}{\partial^2 r}\frac{\partial^2 v}{\partial r^2} + 2w_{11}\frac{\partial^2 u}{\partial r \partial s}\frac{\partial^2 \mathcal{S}}{\partial r \partial s} + w_{02}\frac{\partial^2 u}{\partial s^2}\frac{\partial^2 v}{\partial s^2} dsdr$$

and L_v is a linear form defined as:

$$L_v(u) = -\int_\Omega \nabla P(v) u dr ds.$$

The variational problem:

$$a(\mathcal{S}, u) = L_{\mathcal{S}}(u), \forall u \in H_0^2(\Omega)$$

admits a unique solution if the regularization coefficients (w_{10}, w_{01}, w_{11}, w_{20} et w_{02}) are positive since a is H_0^2-elliptic.

The finite element method gives a discrete approximation v_h of the solution v in a finite dimension space $V_h \subset H_0^2(\Omega)$ verifying:

$$\lim_{h \to 0} \|v - v_h\| = 0.$$

Function v_h is piecewise polynomial on a V_h basis whose generative functions are defined on a restricted domain.

Ω is split into triangular or rectangular elements. Rectangular elements do not allow the representation of surfaces with unrestricted topology but they lead to a smaller linear system that is easier to solve. In the following, we will consider the set of nodes $\Omega_{\Delta r \Delta s} = \{N_{m,n}\}_{m,n}$ such as $N_{m,n} = (m\Delta r, n\Delta s)$. It defines the rectangular finite elements $K_{m,n} = [N_{m,n}, N_{m+1,n}] \times [N_{m,n}, N_{m,n+1}]$.

We consider Bogner-Fox-Schmidt finite elements. Let $Q_3(\mathbb{R}^2)$ be the set of polynomials $\left\{p(r,s) = \sum_{0 \le k,l \le 3} \gamma_{k,l} s^k r^l\right\}$. V_h is defined by:

$$V_h = \left\{v \in \mathcal{C}^1(\Omega) | v_{|K_{i,j}} \in Q_3(K_{i,j})\right\}$$

where $Q_3(K_{i,j}) = \left\{p_{|K_{i,j}} | p \in Q_3(\mathbb{R}^2)\right\}$. Bogner-Fox-Schmidt finite elements basis can be obtained by the tensor product of Φ and Ψ base functions of Hermite finite elements in $\mathbb{R}$:

$$\Phi(x) = \begin{cases} -\frac{2}{h^3}x^3 - \frac{3}{h^2}x^2 + 1 & \text{if } x \in [-h, 0] \\ \frac{2}{h^3}x^3 - \frac{3}{h^2}x^2 + 1 & \text{if } x \in [0, h] \\ 0 & \text{else} \end{cases} \qquad \Psi(x) = \begin{cases} \frac{1}{h^2}x^3 + \frac{2}{h}x^2 + x & \text{if } x \in [-h, 0] \\ \frac{1}{h^2}x^3 - \frac{2}{h}x^2 + x & \text{if } x \in [0, h] \\ 0 & \text{else} \end{cases}$$

Thus v_h is given by:

$$v_h(r,s) = \sum_{|k-i|<1, |l-j|<1} \left(v_h(N_{k,l})\Phi(r) + \frac{\partial v_h}{\partial r}(N_{k,l})\Psi(r)\right)\Phi(s) + \left(\frac{\partial v_h}{\partial s}(N_{k,l})\Phi(r) + \frac{\partial^2 v_h}{\partial r \partial s}(N_{k,l})\Psi(r)\right)\Psi(s)$$

This is a linear system with unknown parameters $v_h(N_{k,l})$, $\frac{\partial v_h}{\partial r}(N_{k,l})$, $\frac{\partial v_h}{\partial s}(N_{k,l})$ and $\frac{\partial^2 v_h}{\partial r \partial s}(N_{k,l})$ to each node. This system may be written:

$$\mathbf{Kx} = \mathbf{f}$$

and is equivalent to equation 11.

As finite elements, B-splines represent the model as a tensor product of polynomial curves. The model shape is controlled by a set of nodes. The B-spline representation is indeed equivalent to finite elements on a particular basis [28].

The finite element method only requires the computation of the surface second derivatives as opposed to finite differences that make use of fourth derivatives. Furthermore, it allows the computation of the surface differential attributes at any point, not only at the nodes. In some finite element methods, v_h is not linear and implies greater order combinations. The finite element method requires fewer nodes than finite differences since it provides an interpolation of the surface between nodes. However, finite elements are computationally expensive compared to finite differences.

3.5 Global minimization

The energy functional is usually non-convex making the global minimization process difficult. Few approaches have been proposed to find the energy global minima.

3.5.1 Simulated annealing

The simulated annealing [107] derives from the gradient descent method. Starting from an initial surface $\mathcal{S}_0$, a gradient descent iteratively evolves according to the equation:

$$\mathcal{S}_{k+1} = \mathcal{S}_k - \Delta k \nabla \mathbf{E}(\mathcal{S}_k)$$

until the model converges. The simulated annealing introduces a random component in the convergence process. Let $\mathbf{q}$ denote the parameter vector of the discretized energy functional $\mathbf{E_q}$. At each iteration, a component q_i from vector $\mathbf{q}$ is randomly selected and the energy variation

$$\Delta \mathbf{E_q} = \mathbf{E}_{\{q_0,\ldots,q_i+\xi,\ldots,q_n\}} - \mathbf{E_q}$$

resulting from q_i variation by a constant ξ is evaluated. If the energy variation is negative, the energy descent is accepted. If $\Delta \mathbf{E}$ is positive, the energy augmentation might be accepted with a probability $e^{\frac{\Delta \mathbf{E_q}}{t}}$ where t is a «temperature» parameter decreasing from an initial value towards 0 as the algorithm converges. A high temperature makes it possible to get out of the energy local minima. As the system «freezes», the algorithm becomes identical to a gradient descent. Due to its stochastic nature and the difficulty to set the temperature evolution law, simulated annealing may be very slow to converge and tricky to optimize.

3.5.2 Dynamic programming

Dynamic programming was introduced by Amini *et al* [3] to solve the evolution equation of deformable contours. It is based on a particular discretization of the contour energy:

$$E(\mathcal{C}) = \sum_{i=0}^{d-1} E_{\text{reg}}(\mathbf{p}_i) + E_{\text{ext}}(\mathbf{p}_i),$$

where $\mathcal{C} = \{\mathbf{p}_0, \ldots, \mathbf{p}_{d-1}\}$. This energy can be written:

$$E(\mathcal{C}) = E_0(\mathbf{p}_0, \mathbf{p}_1) + E_1(\mathbf{p}_1, \mathbf{p}_2) + \ldots + E_{d-1}(\mathbf{p}_{d-1}, \mathbf{p}_{d-2}). \tag{12}$$

We consider the following notations:

$$\begin{cases} s_0(\mathbf{p}_1) = \min_{\mathbf{p}_0} E_0(\mathbf{p}_0, \mathbf{p}_1) \\ \forall k \in [1, d-1], s_k(\mathbf{p}_{k+1}) = \min_{\mathbf{p}_k} \{s_{k-1}(\mathbf{p}_k) + E_k(\mathbf{p}_k, \mathbf{p}_{k-1})\} \end{cases}.$$

A global minimum $\min E = \min_{\mathbf{p}_{d-1}} s_{d-2}(\mathbf{p}_{d-1})$ of the energy functional may be computed by dynamic programming. The contour leading to this minimum is built by tracking back into the position matrix. Geiger *et al* [51] propose an optimization to this method by using a multi-scale approach.

Dynamic programming is one of the rare method finding the energy functional global minimum. It is possible to introduce in the deformation process hard constraints such as a minimal distance between each vertex pair. The algorithm might then fail if no solution satisfy the given hard constraints.

This algorithm extension to the surface case is made difficult by the complex decomposition of the energy functional as a sum equivalent to equation 12.

3.5.3 Other global minimization technics

The global minimization problem may be solved when additional constraints that regularize the energy functional are available. Cohen and Kimmel [31, 30] find the minimal energy path between two fixed points. An iterative algorithm is used to compute the level sets of a potential function from the starting point. This algorithm determines the energy global minimum path.

3.6 Forces equilibrium

Each term of the stationary equation 9 might be seen as an equilibrium between internal forces (the left term) and external forces (the right term) that deform the surface. *Ad-hoc* internal forces f_{int} that enforce some regularizing constraints and external forces f_{ext} that steer the surface deformations towards the data might be computed without considering any energy term [43]. The optimal model then verifies the equilibrium equation:

$$\sum f_{\text{int}}(\mathcal{S}) = -\sum f_{\text{ext}}(\mathcal{S})$$

which can be solved in a similar way to the energy stationary equation 9.

Since these forces do not derive from an energy, it is needed to check that they will lead to the desired smoothing level and to a stable behavior of the deformation process. The internal forces must make the model evolve towards a unique shape and they have to vanish when the regularizing constraints are satisfied. The external forces must make the model converge towards the data without oscillations. External forces proportional to the distance between the data and the model should be used. These forces vanish when the model coincide with the data.

3.7 Evolution equation

Solving the stationary equation requires an iterative method which is initialized with an approximate solution to converge towards a minimum. The first approach is based on a direct minimization technique of a multivariate function such as the gradient descent. Many of these unconstrained minimization schemes involve a line search for minimum along a given direction, and a strategy for choosing a new direction, and essentially differ from the latter strategy. Another approach is to introduce an evolutional term in the stationary equation then to iteratively solve the differential equation.

3.7.1 Solving by a gradient descent

A gradient descent or one of its derived algorithms [95, 72] allow the minimization of a function as a model energy. We suppose in this section that the energy E is discretized using a finite difference scheme as detailed in section 3.3 for instance.

Gradient descent

This method is characterized by the evolution of an initial surface $\mathcal{S}_0$ in the steepest energy direction. The surface iteratively evolves according to the equation:

$$\mathcal{S}_{k+1} = \mathcal{S}_k - \Delta t \nabla \mathbf{E}(\mathcal{S}_k) \tag{13}$$

where Δt is a positive constant. This method converges towards an energy local minimum. The initial surface used is thus meaningful.

When the algorithm reaches a minimum of the energy function in a given direction, then the steepest direction at that point is necessarily orthogonal to the direction previously followed. A gradient descent thus evolves by constant steps with directions orthogonal to each others. This procedure does not follow, in the general case, the shortest path toward a local minimum. Hence, the convergence might require a larger number of iterations.

Conjugate gradient

The conjugate gradient method is similar to the gradient descent method but a different direction is followed at each algorithm iteration. A direction orthogonal to every earlier directions is computed. Let $\mathbf{g}_k = -\nabla \mathbf{E}(\mathcal{S}_k)$. If $\mathbf{d}_k$ is the direction followed by the algorithm

at iteration k, the direction:

$$\mathbf{d}_{k+1} = \mathbf{g}_k + \frac{(\mathbf{g}_k - \mathbf{g}_{k-1}).\mathbf{g}_k}{\mathbf{g}_{k-1}.\mathbf{g}_{k-1}}\mathbf{d}_k$$

is conjugated to directions $\mathbf{d}_i, i \in [0,k]$. If $\mathbf{E}$ is a quadratic form, the algorithm is optimal and requires exactly m iterations, where m is $\mathbf{E}$ matrix dimension.

Newton method

If the energy functional belongs to $\mathcal{C}^2$, it may be approximated by its Taylor expansion up to the second order in the neighborhood of any point $\mathcal{S}_k$:

$$T(\mathcal{S}) = \mathbf{E}(\mathcal{S}_k) + (\mathcal{S} - \mathcal{S}_k)\nabla\mathbf{E}(\mathcal{S}_k) + \frac{1}{2}(\mathcal{S} - \mathcal{S}_k)^T\nabla^2\mathbf{E}(\mathcal{S})(\mathcal{S} - \mathcal{S}_k)$$

If the Hessian matrix $\nabla^2\mathbf{E}(\mathcal{S}_k)$ is positive definite then T admit a unique minimum defined by:

$$\mathcal{S}_{k+1} = \mathcal{S}_k - \left[\nabla^2\mathbf{E}(\mathcal{S}_k)\right]^{-1}\nabla\mathbf{E}(\mathcal{S}_k) \tag{14}$$

In the case of a quadratic form, this method converges towards the minimum in only one iteration. In any other case, it is needed to iterate until convergence.

Levenberg-Marquardt method

If the energy functional is close to a quadratic form, the Newton method (equation 14) is efficient and leads to a good approximation of the result. In the other cases, a gradient descent (equation 13) is most efficient. The Levenberg-Marquardt method continuously go from a Newton convergence to a gradient descent as the process converges. Both equation 13 and equation 14 can be written:

$$\mathbf{A}(\mathcal{S}_k)(\mathcal{S}_{k+1} - \mathcal{S}_k) = \nabla\mathbf{E}(\mathcal{S}_k) \quad \text{with} \quad \mathbf{A}(\mathcal{S}_k) = \begin{cases} \frac{\mathbf{I}}{\Delta t} & \text{for equation } 13 \\ \nabla^2\mathbf{E}(\mathcal{S}_k) & \text{for equation } 14 \end{cases} \tag{15}$$

where I is an identity matrix. The Levenberg-Marquardt method define $\mathbf{A}$ as:

$$\mathbf{A}(\mathcal{S}_k) = \nabla^2\mathbf{E}(\mathcal{S}_k) + \rho\mathbf{I}$$

in equation 15.

If constant ρ is null or almost null, the algorithm evolves as the Newton method. Otherwise, if ρ is large, the diagonal terms of $\mathbf{A}(\mathcal{S}_k)$ matrix prevail and the resolution is close to a gradient descent.

In [124], Vemuri *et al* introduce new numerical algorithms and compare their efficiency.

3.7.2 Evolutive approach

The discrete stationary equation 11 is not linear. Its resolution requires an evolutive framework that iterates towards the solution. Many methods for solving iterative systems are proposed in [95].

The evolutive approach consider the deformable model as a dynamic object evolving according to a parameter t [116]:

$$\begin{array}{rcccl} \mathcal{S} & : & \Omega \times [0,+\infty] & \rightarrow & \mathbb{R}^n \\ & & (\mathbf{u},t) & \mapsto & \mathcal{S}(\mathbf{u},t) \end{array}$$

and the corresponding energy term:

$$\mathbf{E}(\mathcal{S},t) = \mathbf{E}(\mathcal{S}(.,t)).$$

Inertial terms are introduced in equation 11, of the first order (Lagrangian evolution):

$$\frac{\partial \mathcal{S}}{\partial t} + \mathbf{K}(\mathcal{S})\mathcal{S} = \mathbf{f}(\mathcal{S})$$

or of the second order (Newtonian evolution) :

$$\mathbf{M}\frac{\partial^2 \mathcal{S}}{\partial t^2} + \mathbf{C}\frac{\partial \mathcal{S}}{\partial t} + \mathbf{K}\mathcal{S} = \mathbf{f}(\mathcal{S}), \tag{16}$$

where $\mathbf{M}$ and $\mathbf{C}$ are two $NM \times NM$ matrices representing the model mass and the background damping. M et N are the dimensions of the discretized parameter space.

This formulation leads to an evolutional scheme suited to active surfaces since it allows the user to interact with the model during the deformation process. It requires setting an initial surface $\mathcal{S}_0 = \mathcal{S}(.,0)$ close enough to the solution. The initial surface evolves until the inertial terms vanish. The surface then satisfies the stationary equilibrium between regularizing and data forces.

Since the system deformations do not have any physical meaning, authors often consider that the system mass is null thus suppressing the second order term in equation 16 [117, 27].

The dynamic system 16 must be discretized in time by the finite difference scheme. Given a Δt time step, partial derivatives might be approximated by:

$$\frac{\partial \mathcal{S}}{\partial t} \simeq \frac{\mathcal{S}_{t+\Delta t} - \mathcal{S}_{t-\Delta t}}{2\Delta t} \simeq \frac{\mathcal{S}_{t+\Delta t} - \mathcal{S}_t}{\Delta t} \text{ et } \frac{\partial^2 \mathcal{S}}{\partial t^2} \simeq \frac{\mathcal{S}_{t+\Delta t} - 2\mathcal{S}_t + \mathcal{S}_{t-\Delta t}}{\Delta t^2}.$$

$\mathbf{M}$ and $\mathbf{C}$ are diagonal or banded matrices. They lead to independent equations and are easily invertible. The time discretization leads either to an explicit numerical scheme, or to an implicit numerical scheme.

An explicit numerical scheme:

$$\mathbf{M}\frac{\mathcal{S}_{t+\Delta t} - 2\mathcal{S}_t + \mathcal{S}_{t-\Delta t}}{\Delta t^2} + \mathbf{C}\frac{\mathcal{S}_{t+\Delta t} - \mathcal{S}_t}{\Delta t} + \mathbf{K}\mathcal{S}_t = \mathbf{f}(\mathcal{S}_t), \tag{17}$$

converges only if the selected time step is small enough [95]. However, equation 17 give an explicit way of computing $\mathcal{S}_{t+\Delta t}$ from $\mathcal{S}_t$ and $\mathcal{S}_{t-\Delta t}$:

$$\mathcal{S}_{t+\Delta t} = \left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{\Delta t}\right)^{-1} \left(\left(\frac{2\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{\Delta t} - \mathbf{K}\right)\mathcal{S}_t - \frac{\mathbf{M}}{\Delta t^2}\mathcal{S}_{t-\Delta t} + \mathbf{f}(\mathcal{S}_t)\right). \tag{18}$$

This equation is simple and can be solved at low cost. If $\mathbf{M} = 0$ and $\mathbf{C} = \mathbf{I}$, equation 18 simplifies:

$$\mathcal{S}_{t+\Delta t} = \mathcal{S}_t - \Delta t(\mathbf{K}\mathcal{S}_t - \mathbf{f}(\mathcal{S}_t)) = \mathcal{S}_t - \Delta t \nabla \mathbf{E}.$$

It can be seen that the Lagrangian explicit numerical scheme leads to an equation identical to a gradient descent 13.

An implicit numerical scheme:

$$\mathbf{M}\frac{\mathcal{S}_{t+\Delta t} - 2\mathcal{S}_t + \mathcal{S}_{t-\Delta t}}{\Delta t^2} + \mathbf{C}\frac{\mathcal{S}_{t+\Delta t} - \mathcal{S}_{t-\Delta t}}{2\Delta t} + \mathbf{K}\mathcal{S}_{t+\Delta t} = \mathbf{f}(\mathcal{S}_{t+\Delta t}).$$

cannot be solved since it would require the knowledge of $\mathbf{f}(\mathcal{S}_{t+\Delta t})$. In practice, a semi-implicit scheme is used:

$$\mathbf{M}\frac{\mathcal{S}_{t+\Delta t} - 2\mathcal{S}_t + \mathcal{S}_{t-\Delta t}}{\Delta t^2} + \mathbf{C}\frac{\mathcal{S}_{t+\Delta t} - \mathcal{S}_{t-\Delta t}}{2\Delta t} + \mathbf{K}\mathcal{S}_{t+\Delta t} = \mathbf{f}(\mathcal{S}_t). \tag{19}$$

The semi-implicit scheme is more stable than the explicit scheme and insures the numerical convergence even for larger time steps. However, the semi-implicit scheme 19 requires the resolution of a dependent equation set at each iteration:

$$\mathcal{S}_{t+\Delta t} = \left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{2\Delta t} + \mathbf{K}\right)^{-1}\left(\frac{2\mathbf{M}}{\Delta t^2}\mathcal{S}_t + \left(\frac{\mathbf{C}}{2\Delta t} - \frac{\mathbf{M}}{\Delta t^2}\right)\mathcal{S}_{t-\Delta t} + \mathbf{f}(\mathcal{S}_t).\right)$$

Although a larger time step reduces the number of iterations, the semi-implicit scheme requires the costly inversion of matrix $\mathbf{K}$. This matrix inversion is computed each time the model topology or its physical coefficients w_{ij} change.

Case of parameterized models

In the case of parameterized models, a parameter vector $\mathbf{q}$ defines surface $\mathcal{S}_\mathbf{q} = \mathbf{T}\mathbf{q}$ where $\mathbf{T}$ is a transformation matrix. This discrete representation leads to an evolution equation similar to equation 16:

$$\mathbf{M}'\frac{d^2\mathbf{q}}{dt^2} + \mathbf{C}'\frac{d\mathbf{q}}{dt} + \mathbf{K}'\mathbf{q} = \mathbf{f}',$$

where $\mathbf{M}'$, $\mathbf{C}'$ and $\mathbf{K}'$ depends on $\mathbf{M}$, $\mathbf{C}$, $\mathbf{K}$ and $\mathbf{T}$. The resolution of the dynamic equation is similar to 16.

3.8 Discrete model evolution

A discrete model geometry is defined only at its vertices. Model deformations are not governed by a partial differential equation but by the model vertex displacements compared to its neighbors. A vertex i might be considered as a point mass governed by a Newtonian law of motion:

$$m(i)\frac{d^2\mathbf{p}_i}{dt^2} = -\gamma(i)\frac{d\mathbf{p}_i}{dt} + \alpha(i)f_{\text{int}}(\mathbf{p}_i) + \beta(i)f_{\text{ext}}(\mathbf{p}_i), \tag{20}$$

where $m(i)$ is the vertex mass, $\gamma(i)$ is the background damping, f_{int} is the regularizing force, f_{ext} is the data force, and α and β are two weights controlling the internal and external terms. This equation corresponds to an equilibrium between the acceleration, the inertial force, the regularizing force and the data force.

For a model with n vertices, there are n independent linear equations that may be written with a matrix form equivalent to equation 16:

$$\mathbf{M}\frac{d^2\mathcal{S}}{dt^2} + \mathbf{C}\frac{d\mathcal{S}}{dt} + \mathbf{f}_{\text{int}} = -\mathbf{f}_{\text{ext}},$$

where $\mathbf{M} = \text{diag}(m(0), \ldots, m(n-1))$, $\mathbf{C} = \text{diag}(\gamma(0), \ldots, \gamma(n-1))$, and $\mathbf{f}_{\text{int}}$ and $\mathbf{f}_{\text{ext}}$ are the internal and external force vectors. Solving equation 20 requires a discretization in time. Using centered finite differences and an explicit scheme it leads to [42]:

$$\mathbf{p}_i^{t+\Delta t} = \mathbf{p}_i^t + \left(1 - \frac{\gamma(i)\Delta t}{m(i)}\right)(\mathbf{p}_i^t - \mathbf{p}_i^{t-\Delta t}) + \frac{\Delta t^2}{m(i)}\left(\alpha(i) f_{\text{int}}(\mathbf{p}_i^t) + \beta(i) f_{\text{ext}}(\mathbf{p}_i^t)\right). \quad (21)$$

3.9 Implicit models evolution

Algebraic surfaces

Implicit models are characterized by an equation such as $F_{\mathbf{q}}(\mathbf{p}) = 1$. The distance from a point $\mathbf{p}$ to the surface may be roughly approximated as $F_{\mathbf{q}}(\mathbf{p}) - 1$. Given d data points $\{\mathbf{p}_i\}_{i\in[0,d-1]}$, the model energy may be estimated by:

$$E_{\mathbf{q}}(\{\mathbf{p}_i\}) = \sum_{i=0}^{d-1} |1 - F_{\mathbf{q}}(\mathbf{p}_i)|^2$$

The resolution consist in finding the implicit surface parameters $\mathbf{q}_0$ verifying:

$$E_{\mathbf{q}_0} = \min_{\mathbf{q}} E_{\mathbf{q}}.$$

This energy is minimized by using one of the gradient descent method proposed in section 3.7.1. In many cases, function $|1 - F_{\mathbf{q}}|$ is a poor estimation of the distance from a point to the surface. Authors usually propose better approximations involving a specific knowledge of the surface [97, 114].

Level-set methods

Malladi *et al* [74] qualify as *Eulerian dynamic* the level-set evolution equations by opposition to the *Lagrangian dynamic* of «classical» deformable models. Let $\{\mathcal{S}(\mathbf{u}, t)\}_t$ be a surface family evolving according to the partial differential equation:

$$\frac{\partial S}{\partial t} = \nu\mathbf{n}$$

where $\mathbf{n}$ represents the surface normal and ν the displacement speed. For instance, $\nu = H_{\mathcal{S}}$ the surface mean curvature. Since Ψ is null on $\mathcal{S}$:

$$\Psi(\mathcal{S}(t), t) = 0, \forall t$$

Knowing that $\mathbf{n} = \frac{\nabla\Psi}{|\nabla\Psi|}$ and by differentiation, the previous equation leads to an the Hamilton-Jacobi equation:

$$\frac{\partial\Psi}{\partial t} = \bar{\nu}|\nabla\Psi| \tag{22}$$

where $\bar{\nu}$ is a function of $\mathbb{R}^4 \rightarrow \mathbb{R}$ which coincide with ν on its definition domain. For instance, $\bar{\nu}(\mathbf{u}, t) = H_{\Psi}(\mathbf{u}, t)$, the mean curvature of the level-set crossing $\mathbf{u}$, that coincide with ν on $\mathcal{S}$ ($H_{\Psi}(\mathbf{u}, 0) = H_{\mathcal{S}}(\mathbf{u})$).

If function $\bar{\nu}$ is regular enough, Ψ does not show any singularity as it evolves and equation 22 can be solved by using finite differences (in space and time):

$$\mathbf{\Psi}_{ijk}^{t+\Delta t} = \mathbf{\Psi}_{ijk}^{t} + \Delta t \bar{\nu}_{ijk} |\nabla_{ijk} \mathbf{\Psi}_{ijk}^{t}| \tag{23}$$

Moreover, the surface geometrical attributes are determined by the analytical expressions:

$$\mathbf{n} = \frac{\nabla\Psi}{|\nabla\Psi|} \text{ and } H = \nabla \cdot \frac{\nabla\Psi}{|\nabla\Psi|}$$

For image segmentation, the surface evolution speed ν must make the front stop on object contours. Depending on the properties of ν, it may be difficult to completely stop the propagation.

As the propagation speed may become discontinuous at points far away from the 0 level-set, it is needed to periodically reinitialize Ψ as the distance function to the current surface to ensure evenly spaced level-sets.

Equation 23 is computationally expensive since a 3 dimensional grid must be updated to compute a surface evolution. The computation time might be reduced by computing Ψ value only on a narrow band around the surface $\{\Psi = 0\}$. It is then required to periodically reset the narrow band location in order to avoid collisions of the evolving surface with the band borders.

4 Constraining deformations

A deformable surface evolving according to an evolution law such as equation 16 or 20 is only constrained by the number of degrees of freedom (DOF) imposed by its geometric representation. Some representations have few DOF such as superquadrics with only 6 parameters and may constrain the deformation process too much for a valid reconstruction of complex structures. Conversely, a surface with too many degrees of freedom is more sensitive to noise and outliers. Moreover, most minimization technics only lead to sub-optimal solutions corresponding to energy local minima. The more DOF a surface has, the

less convex its energy functional is, and the more it is likely to converge towards a local minimum.

Therefore, several approaches have been proposed to restrict the model deformation space. These deformation restrictions are often based on a coarse to fine strategy from a highly constrained deformation space towards more local deformations as the model converges towards its optimal shape.

4.1 Using global transformations

A model deformation may be driven by iteratively applying global transformations of the space embedding the model instead of applying local deformations of the model inside space. As discussed in section 2.1, this approach is referred to as *registration* opposed to model local *deformations*. The registration framework restricts the model variation space depending on the choice of global transformations.

Iterative Closest Point algorithm

The registration process described in Besl and McKay [12] and Zhang [131] is very close to the iterative resolution of the deformable models dynamic equation. A model is iteratively deformed by a global transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. The transformation T is estimated by minimizing the least square criterion:

$$T = \arg \min_{T \in \mathrm{T_{reg}}} \sum_{i=0}^{d-1} \|T(\mathbf{p}_i) - (\mathrm{Closest}(\mathbf{p}_i))\|^2 \tag{24}$$

with «Closest($\mathbf{p}_i$)» a data point close to vertex i, and $\mathrm{T_{reg}}$ a given transformation group. The authors refer to the *Iterative Closest Point* algorithm (ICP).

Rigid and affine registration

Widely used transformation groups for T estimation include rigid transformations (6 DOF), similarities (7 DOF) and affine transformations (12 DOF). For these three transformation groups, there exists a closed form solution for solving equation 24 [92]. Rigid registration is useful for aligning data coming from a common source. Similarity and affine registrations are used to align different data in the same reference frame. However, in this case, the small number of DOF usually does not allow the reconstruction of structures from a model with a different prior shape.

Free-form deformations and B-spline transformations

Sederberg and Parry [100] and Coquillart [34] introduce free-form deformations (FFD) where the embedding space is deformed according to a set of control points and interpolation functions between those points such as Bernstein trivariate polynomials. Thus deformations apply locally to each node point and the number of DOF is tuned through the number of control points.

B-spline transformations are a particular case of FFD where the interpolation functions are B-splines. A B-spline curve is defined in 1D as a piecewise polynomial curve. Let m

and n be two integers and $(t_0, t_1, \ldots, t_{n+m})$ be an ordered reals sequence representing the transformation nodes. A n order B-spline function is defined recursively:

$$\forall i \in [0, n-1], \left\{ \begin{array}{l} B_i^0(x) = \left\{ \begin{array}{ll} 1 & \text{if } t_i \leq x \leq t_{i+1} \\ 0 & \text{else} \end{array} \right. \\ B_i^m(x) = \frac{x - t_i}{t_{i+m} - t_i} B_i^{m-1}(x) + \frac{t_{i+m+1} - x}{t_{i+m+1} - t_{i+1}} B_{i+1}^{m-1}(x). \end{array} \right.$$

A volume B-spline transformation is defined by a tensorial product of B-spline curves. Let $\{\mathbf{c}^{ijk}\}_{ijk}$ be the set of control points in $\mathbb{R}^3$ and $\{B_{x,i}^m\}_i$, $\{B_{y,j}^m\}_j$, and $\{B_{z,k}^m\}_k$ be 3 set of 1D B-spline functions on coordinates x, y and z. The B-spline transformation of a point $\mathbf{p}$ writes:

$$T(\mathbf{p}) = \left\{ \begin{array}{l} T_x(\mathbf{p}) = \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sum_{k=0}^{n_z-1} c_x^{ijk} B_{i,x}^m(x) B_{j,y}^m(y) B_{k,z}^m(z) \\ T_y(\mathbf{p}) = \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sum_{k=0}^{n_z-1} c_y^{ijk} B_{i,x}^m(x) B_{j,y}^m(y) B_{k,z}^m(z) \\ T_z(\mathbf{p}) = \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sum_{k=0}^{n_z-1} c_z^{ijk} B_{i,x}^m(x) B_{j,y}^m(y) B_{k,z}^m(z). \end{array} \right.$$

The resolution of equation 24 with B-spline transformations can be achieved iteratively by a gradient descent.

Declerck *et al* [36] use B-spline transformations to warp space. Due to the amount of deformations allowed by B-splines, he introduces an additive Tikhonov stabilizer whose influence in the minimization process is weighted. Lötjönen *et al* [71, 70] and Bardinet *et al* [10, 8] similarly need to add a regularizing term to FFD.

Superquadric extensions

Terzopoulos *et al* [117] introduce a model composed of a reference superquadric on which an additional deformation term allows the definition of local surface deformations. Let $\mathcal{Q}_{\mathbf{q}_l}$ denote a superquadric with parameters $\mathbf{q}_l$. The model surface is defined by:

$$\mathcal{S}_{\mathbf{q}}(r, s) = \mathbf{c} + \mathbf{R} \left(\mathcal{Q}_{\mathbf{q}_l}(r, s) + \mathbf{d}(r, s) \right),$$

where $\mathbf{c}$ represents the $\mathcal{Q}$ superquadric inertial center, $\mathbf{R}$ is a rotation matrix, and $\mathbf{d}$ is a vectorial displacement field from the superquadric surface. The authors define $\mathbf{d}$ as a linear combination of basis functions: $\mathbf{d} = \mathbf{S}\mathbf{q}_d$ where $\mathbf{S}$ is the basis function matrix and $\mathbf{q}_d$ is a weight vector. If $\mathbf{q}_c$ and $\mathbf{q}_\theta$ denote the translation and the rotation parameters respectively, the parameters vector is $\mathbf{q} = (\mathbf{q}_c, \mathbf{q}_\theta, \mathbf{q}_l, \mathbf{q}_d)^T$. This representation leads to more shape variability and can represent a wider set of surfaces than superquadrics.

Metaxas *et al* [81] still introduce more deformations on this model to represent physical deformations of the surface (bending, pinch, twist...). The model surface is defined by :

$$\mathcal{S}_{\mathbf{q}}(r, s) = \mathbf{c} + \mathbf{R} \left(T\mathcal{Q}_{\mathbf{q}_l}(r, s) + \mathbf{d}(r, s) \right),$$

where T is a transformation. The surface parameter extends to $\mathbf{q} = (\mathbf{q}_c, \mathbf{q}_\theta, \mathbf{q}_l, \mathbf{q}_d, \mathbf{q}_T)$.

Other globally constrained deformations

Feldmar and Ayache [50, 49] define locally affine transformations to increase the number of DOF. They determine the «closest» point of a vertex $\mathbf{p}_i$ by computing a distance function

from the data points. Their distance include geometric position of points but also their similarity in terms of normal orientation and principal curvature values. They estimate several affine transformations by considering only a restricted set of vertices included in a given neighborhood around each model vertex. They apply a piecewise affine transformation and demonstrate some continuity results on the surface patches.

Montagnat and Delingette [84] regularize deformations through a modified deformation scheme that combines local and global forces resulting from a global transformation estimation. A global transformation in a given transformation group is first estimated using a least square criterion. The «closest» point from a surface vertex is estimated using the external force field. The global transformation results in a global force field that applies onto the surface model. By weighting the global and the local forces, it is possible to continuously evolve from a global registration process to a local deformation one.

4.2 Coarse-to-fine approaches

4.2.1 Energy Graduated Non-Convexity

The energy functional is usually non-convex making the deformation process very dependent of the model initialization. The Graduated Non-Convexity algorithm (GNC) proposed by Blake and Zisserman [15] makes the energy functional locally convex. The idea is to make the (positive) second derivative of the energy regularizing term greater than the (negative) second derivative of the energy data term to produce a convex approximation E^1 of the energy functional E. E^1's minimum does not necessarily coincide with E's global minimum. An energy family $\{E^i\}_{i\in[0,1]}$ is built such as E^1 is convex and $E^0 = E$. By varying i from 1 to 0 and making the model iteratively evolve according to energy E^i, it gets closer from the local minimum of E. Blake and Zisserman demonstrate the validity of the GNC algorithm in the particular case of their study but they state that it is not guaranteed to converge in the general case.

Davatzikos and Prince [35] similarly build a functional family varying with a scalar parameter to drive a thick line detector model. Cohen and Gorre [29] use the GNC algorithm for deformable modelling in the case where the data term is a function of the distance from a point $\mathbf{p}$ to the closest image boundary point.

4.2.2 Pyramidal approach

Reissman *et al* [96] model an image with a graph of cells describing the image regions. A pyramid is build from the original graph by merging graph regions from one pyramid layer to the above one. The cells are deformed accordingly with an energy function. The algorithm proceeds to a coarse to fine deformation by deforming first the pyramid apex and then propagating the computed deformations towards the base. Lötjönen *et al* [71] uses a similar idea with deformable surfaces. They build a multi-resolution surface model by decimation of an original mesh and the mesh is deformed in a corresponding image pyramid from the apex towards the base.

Szeliski [110] also proposes a hierarchical approach to minimize an energy functional. He builds a multi-scale pyramid of basis functions. It is then possible to compute the discrete energy at different precision levels. The energy minimization begins at a coarse scale which is refined with the algorithm convergence.

4.3 Local regularization constraints extension

To enforce the regularizing behavior of the energy internal term, Feldmar and Ayache [50] compute the internal term over a neighborhood with a variable size at each model point. The computation neighborhood is geometrically defined at each mesh point by the set of mesh points included in a sphere of radius r. The larger r is, the stronger the regularization term is. Montagnat and Delingette [83] similarly use a topological neighborhood and show the regularizing effect on surface model deformations.

4.4 Modal analysis

4.4.1 Surface vibration modes

Decomposition of a surface into a set of modes provides a coarse to fine representation by tuning the number of modes used. The vibration modes, extracted by Fourier analysis for example (see section 2.3 and [105]), give an incrementally detailed representation of surfaces.

Pentland *et al* [93] similarly optimize the resolution of the model evolution 16 by decomposing the surface $\mathcal{S}$ on a vibration basis. Let $\mathcal{S}$ be a n dimensional vector and $\tilde{\mathcal{S}} = \mathbf{\Phi}\mathcal{S}$ where $\mathbf{\Phi}$ is an orthogonal transformation matrix. Equation 16 rewrites:

$$\tilde{M}\frac{d^2\tilde{\mathcal{S}}}{dt^2} + \tilde{C}\frac{d\tilde{\mathcal{S}}}{dt} + \tilde{K}\tilde{\mathcal{S}} = \tilde{F}, \tag{25}$$

where $\tilde{\mathbf{M}}$, $\tilde{\mathbf{C}}$, $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{f}}$ depend on $\mathbf{M}$, $\mathbf{C}$, $\mathbf{K}$, $\mathbf{f}$ and $\mathbf{\Phi}$.

An optimal transformation matrix Φ is the common eigenmodes matrix of M, K and C such that components of equation of 25 are independent. Φ is the solution to the double eigenproblem:

$$\mathbf{K\Phi} = \mathbf{\Omega}^2\mathbf{\Phi M},$$

where $\mathbf{\Phi} = [\phi_1, \phi_2, \ldots, \phi_{3n}]$ and $\mathbf{\Omega} = \mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_{3n})$ are such that the eigenvectors are M-orthonormal, which means that $\mathbf{\Phi}^T\mathbf{K\Phi} = \mathbf{\Omega}^2$ and $\mathbf{\Phi}^T\mathbf{M\Phi} = \mathbf{I}$. The eigenvector ϕ_i is the i^{th} shape vector and ω_i is the associated vibration frequency.

Finding the eigenmodes of the damping matrix $\mathbf{C}$ is possible if it is defined as a Caughey sequence:

$$\mathbf{C} = \mathbf{M}\sum_{k=0}^{p-1} a_k[\mathbf{M}^{-1}\mathbf{K}]^k.$$

If $p = 2$ then $\mathbf{C} = a_0\mathbf{M} + a_1\mathbf{K}$ and equation 25 simplifies:

$$\frac{d^2\tilde{\mathcal{S}}}{dt^2} + \tilde{\mathbf{C}}\frac{d\tilde{\mathcal{S}}}{dt} + \mathbf{\Omega}^2\tilde{\mathcal{S}} = \mathbf{\Phi}^T\tilde{\mathbf{f}},$$

where $\tilde{\mathbf{C}} = a_o\mathbf{I} + a_1\mathbf{\Omega}^2$. This equation system is made of $3n$ independent equations that can be resolved by discretizing the time.

Nastar and Ayache [89] use Pentland and Sclaroff's modal analysis to reduce the model deformation space. Bookstein [16] similarly decomposes a thin-plate spline deformations over the eigenmodes of the bending energy matrix. Cohen and Cohen [26] propose an alternative approach with an hyperquadric based model. The model is defined as a sum of an hyperquadric generating a convex envelop and a number of hyperquadric exponential allowing local non-convex deformations. The model energy is controlled simultaneously by the hyperquadric and the additional term parameters. By iteratively adding terms, they produce a hierarchical deformation scheme.

4.4.2 Superquadric-based modal analysis

Starting from Terzopoulos *et al* superquadric extension [117], Vemuri and Radisavljevic [125] propose a continuous transition between global deformations (only involving variations of the superquadric parameters) and local deformations (involving the $\mathbf{d}$ displacement field). They project $\mathbf{q}_d$ on a wavelet orthogonal basis and set the number of meaningful deformation modes. The more deformation modes are used, the more local deformations apply on the superquadric surface. By limiting the number of modes involved in the deformation computation, they restrict the deformation space. They gradually increase the number of modes as the model converges.

4.4.3 Principal Component Analysis

Cootes *et al* [32] use *principal component analysis* (PCA) (also known as Karhunen-Loeve decomposition) to decompose the model in eigenmodes obtained from a shape training set. The decomposition basis size depends on the training set number of shapes. Since only shapes composed by a linear combination of the eigenmodes can be represented, PCA restricts the shape variations of the model close to the known statistical shape variation.

PCA modelling

PCA only takes into account local deformations after training shapes have been registered by translations, rotations and scale factors. A particular shape is defined by:

$$\mathcal{S}_i = T(\mathbf{x}_i)$$

where $\mathbf{x}_i$ is a parameter vector of dimension d that describes the surface $\mathcal{S}_i$. Vector $\mathbf{x}_i$ may be a set of $d/3$ vertices coordinates that represent a discrete model: $\mathbf{x}_i = (x_1, \ldots, x_d)^T$ [7, 33]. In that case, T is a similarity transformation. Vector $\mathbf{x}_i$ may also be a parameter vector: $\mathbf{x}_i = \mathbf{q} = (q_1, \ldots, q_d)^T$ [113]. In that case, T is a transformation from the parameters space to the shape deformation space. In both cases, let us consider a training set built from n shape instances: $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. The mean shape is defined by:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

and the covariance matrix (positive definite) is:

$$\mathbf{C} = \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

By diagonalization of $\mathbf{C}$, $\mathbf{CU} = \mathbf{UV}$ where $\mathbf{U}$ is the matrix for which each column is an eigenvector of $\mathbf{C}$ and $\mathbf{V} = \text{diag}(v_1, \ldots, v_n)$ is the diagonal matrix of associated eigenvalues. The column vectors, $\mathbf{u}_i$, of matrix $\mathbf{U}$ correspond to «meaningful» deformations according to the training set. PCA is based on the hypothesis that data distribution is Gaussian which leads to a linear representation. Vector $\mathbf{x}_i$ varies inside an n dimensional hyperellipsoid. The eigenvectors of $\mathbf{C}$ represent the hyperellipsoid axis directions while the eigenvalues are the axis amplitudes.

A new shape instance is defined as a linear combination of the eigenmodes:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Uq}$$

where $\mathbf{q}$ is the parameter vector weighting the contribution of each deformation mode. It is generally wanted to prevent the shape to vary too much from the mean shape. Each component q_i of $\mathbf{q}$ is thus restricted to an interval: $-kv_i \leq q_i \leq kv_i$, with $k \in I\!N$.

Usually, n is much lower than d and diagonalization of matrix $\mathbf{C}$ leads to $n-1$ eigenvectors associated to $n-1$ non zero eigenvalues. The number of usable modes depends on the size of the training set. It must be large enough compared to the parameter vector size d to allow the model to recover the 3D object. This approach requires as a preliminary the construction of a training set. This implies to define on a set of n shape instances d matching points. Often considered as a first manual stage, this task is fastidious and may become impracticable in the 3D case. Indeed, to determine corresponding points on smooth surfaces is a difficult task [23] for a human operator except for a few feature points such as curvature extrema. Anyway, several methods to automatically match different shapes are proposed by Caunce and Taylor [19].

PCA constrained deformations

If $\mathbf{x}$ is a model instance in the centered space ($\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pq}$), the surface is a function of $\mathbf{x}$:

$$\mathcal{S} = \mathbf{ERx} + \mathbf{t},$$

where $\mathbf{E} = \text{diag}(e, e, \ldots, e)$ is a scale factor matrix, $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is a translation vector.

Cootes *et al* [33] use a discrete model for which displacement $d\mathcal{S}$ are computed at each iteration. A time $t + \Delta t$ a model instance $\mathbf{x}_{t+\Delta t}$ verifies:

$$\mathcal{S}_t + d\mathcal{S} \simeq \mathbf{E}_{t+\Delta t}\mathbf{R}_{t+\Delta t}\mathbf{x}_{t+\Delta t} + \mathbf{t}_{t+\Delta t}$$

leading to, with $\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \mathbf{dx}$:

$$\mathbf{dx} = \mathbf{E}^{-1}_{t+\Delta t}\mathbf{R}^{-1}_{t+\Delta t}(\mathcal{S}_t + d\mathcal{S} - \mathbf{t}_{t+\Delta t}) - \mathbf{x}_t.$$

To estimate the displacement $\mathbf{dx}$, the eigenmode parameters leading to a model of like shape have to be estimated. In the general case, the eigenmode restrictions prevent from reconstructing exactly $\mathbf{x} + \mathbf{dx}$. A least mean square approximation is obtained by:

$$\mathbf{x} + \mathbf{dx} \simeq \bar{\mathbf{x}} + \mathbf{P}(\mathbf{q} + \mathbf{dq}) \Rightarrow \mathbf{dq} = \mathbf{P}^T \mathbf{dx}.$$

Vector $\mathbf{dq}$ is truncated according to the covariance matrix eigenvalues to prevent the excessive deformation in a given mode direction ($-kv_i \leq q_i + dq_i \leq kv_i$), where k is a positive constant.

PCA encountered a large success for segmentation [33], in particular segmentation of anatomical structures [99, 57], contour tracking [56, 62], and shape classification [90]. Some non linear extensions have been proposed in [20, 104, 103].

5 Deformable models topology

Deformable models give prior information on the shape of the objects to recover. However, if the prior model shape is too different from the data, the model might not be able to deform correctly. Several authors propose topology adaptive models to face this problem.

The term topology change may include both change in the mesh structure (as illustrated in top row of figure 6) or change of the surface genus (bottom row of figure 6) which transforms the surface such as it is not homeomorphic to its previous configuration. In the first case we shall refer to model *adaptation* while genus changes apply to *variable topology* models.

5.1 Adaptive models

Model adaptation is needed to obtain a geometric representation with a detail level similar to the data precision.

Explicit models adaptation

Parameterized models require a data adaptation if the parameter space discretization is rough compared to the data detail level. McInerney and Terzopoulos [76] propose to resample the finite element method nodes along time. This multi-scale approach allows a transition from coarse to fine level. The model is first roughly discretized, and its surface is adapted as it converges towards the data. This approach considerably reduces the computational task at the beginning of the deformation process.

In the case of explicit models, regular sampling of the parameters space does not necessarily lead to a regular sampling of the surface. Vemuri and Radisavljevic [126] propose a transformation of parameter space solving the problem in their particular case. Székely *et al* [113] search to keep constant the ratio between a surface triangle area and the curved triangle area composed with the three same parameters on a unit sphere.

From a numerical point of view, parameterized models resampling necessarily lead to a new computation of mass, damping and stiffness matrices. In the case of a semi-implicit

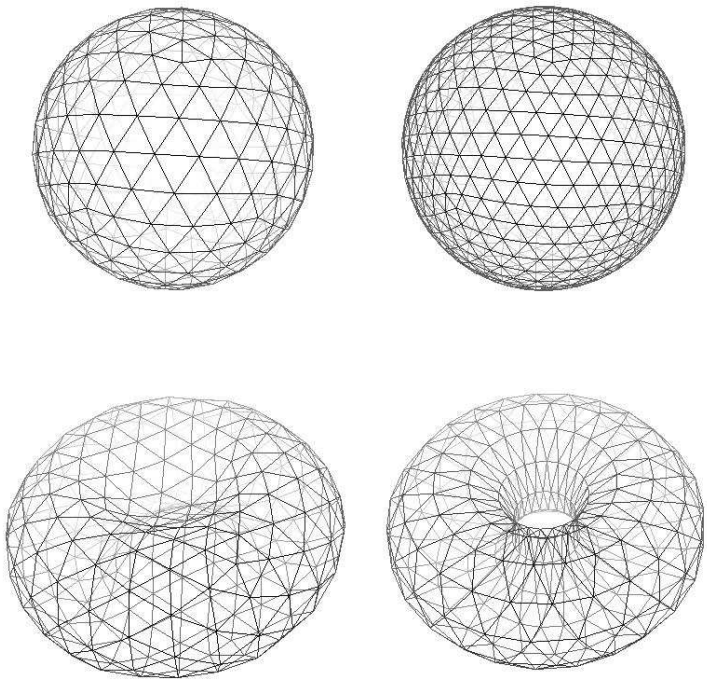

Figure 6: Top row: adaptive topology. Bottom row: topology change.

numerical scheme, these matrices have to be inverted. An adaptive scheme with too high a frequency might thus become computationally expensive.

Discrete adaptive models

In the case of discrete models, adaptation is performed locally by adding new vertices. Since the motion equations are local to each vertex, the surface adaptation has a reduced effect on the evolution scheme. Bulpitt and Efford [18] propose a refinement algorithm for regular triangulated surfaces (each vertex has six neighbors). Delingette [39] defines adaptation laws preserving the simplex meshes geometry.

Subdivision surfaces allow a multi-scale representation of discrete meshes, regardless of their structure. A subdivision surface is defined as a point set that can be enlarged to refine the surface.

The most common approach is to consider the point set $\left\{\mathbf{p}_i^0\right\}_i$ as vertices of a mesh $\mathcal{M}^0$. A mesh sequence is built $(\mathcal{M}^0, \ldots, \mathcal{M}^j, \ldots)$ which produces more and more detailed meshes and converges towards a continuous surface. Thus, $\lim_{j\to\infty} \mathbf{p}_i^j$ exists for any point i and the sequence $\left\{\mathbf{p}_i^j\right\}_j$ strongly converges.

A large number of subdivision surfaces refinement schemes have been proposed. They attempt to preserve the surface properties:

- *Interpolation.* New vertices belong to the continuous surface.
- *Locality.* The neighborhood involved in a new vertex location computation should be as small as possible.
- *Symmetry.* The subdivision scheme has the same symmetry as the local mesh topology.
- *Globality.* The scheme should apply to any discrete surface regardless its topology, even in the presence of mesh borders.
- *Smoothness.* The produced surfaces should enforce a given continuity level.

It is difficult to optimize every criterion at the same time. Several surface subdivision schemes are described in [132].

A second approach is to consider the points $\left\{\mathbf{p}_i^0\right\}_i$ as a control node set indirectly generating a surface. The surface is refined by adding control points.

Adapting the mode number

Decomposing a model on a vibration mode basis as described in [87, 88, 126, 93] leads to an adaptive representation. A restricted set of low frequency modes give a rough surface description. By adding modes, the surface representation is refined.

5.2 Adaptive topology

Several authors propose deformable models whose topology may change along the deformation process to fit the data topology [64, 79, 65]. The model shape initialization may

represent the data very roughly. The reconstruction process does not rely on the prior model shape and benefits of a large number of degrees of freedom.

If the idea of an automatic topology adaptation is attractive in theory, it is difficult to apply when considering noisy images. The deformable models initialization is very dependent on its prior shape and location. To automatically determine topology changes in a noisy image without prior shape information may be unstable.

Parameterized representations

When considering parameterized models, different topologies lead to different parameter boundary conditions. Leitner and Cinquin [66] propose an algorithm to detect self-intersections of surface models represented as a tensor product of B-splines. Surface self-intersections are treated by refining the model, making holes around intersecting regions, and connecting by reordering the control points defining the surface.

Implicit representation

Level-set may change their topology in a very natural way. As the function Ψ continuously evolves, the isovalues define surfaces that may undergo fusions or splitting while Ψ remains a function.

Discrete representation

McInerney and Terzopoulos [77] propose deformable discrete contours («*T-snakes*») and deformable discrete surfaces («*T-surfaces*») with an adaptive topology. Their approach is based on a simplicial decomposition of space using a quasi regular tetrahedral grid. The model is initially a triangulation whose vertices are located on the simplicial grid edges (see figure 7 for an illustration in the 2D case). The algorithm alternatively performs deformations and resampling:

- Deform the model using a Lagrangian dynamic law on discrete meshes for a few steps.
- Approximate the deformed model by a *T-surface* by resampling the triangulation on the simplicial grid.

While the surface is deforming, the vertices of the simplicial grid inside and outside the surface are labelled. After the deformation stage, the surface is redefined as the interface between the inner and the outer vertices. A new triangulation is resampled on the simplicial grid that approximates the deformed model. If the model self intersects or if two models intersect, one of the simplicial grid tetrahedron is shared by two different surface triangles. Topology changes then occur and the colliding triangles merge.

This approach is efficient due to the cheap resampling procedure of the discrete model. However, it requires the definition of the inside and the outside part of the surface. Thus it is not possible to deal with open contours or surfaces with borders. The deformation process brings the surface to inflate or to deflate everywhere. It may cause some problem unless the surface is carefully initialized.

Lachaud and Montanvert [64] define topology adaptive triangulations based on the δ-snakes concept [6]. A length parameter ($\delta > 0$) is used to control the triangulation sampling and to detect model self-intersections from two elementary rules:

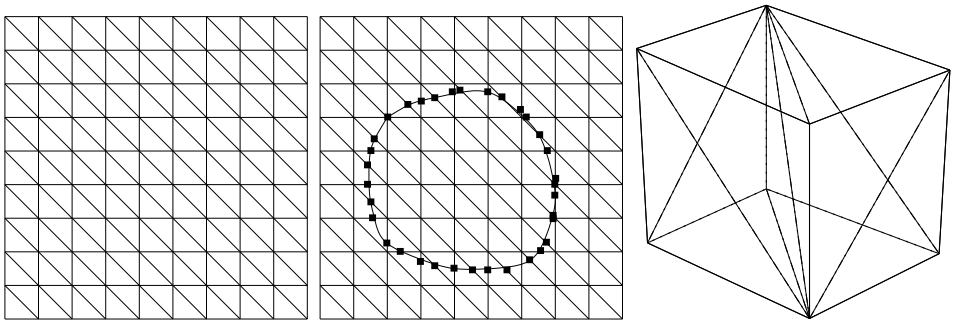

Figure 7: Left : triangular simplicial decomposition of $\mathbb{R}^2$. Center : T-snake. Right : simplicial cell of $\mathbb{R}^3$.

- The distance between two neighbor vertices should lie between δ and 2.5δ.
- The distance between two non neighbor vertices should be greater than $\frac{2.5}{\sqrt{3}}\delta$.

The first constraint enforces a regular sampling of the surface while the second one set a threshold above which the surface is considered to self intersect. If the first constraint fails, a topological operator for creation, destruction or vertices swap is applied (see figure 8). If the second constraint fails, an axial or an annular transformation is applied (see figure 9).

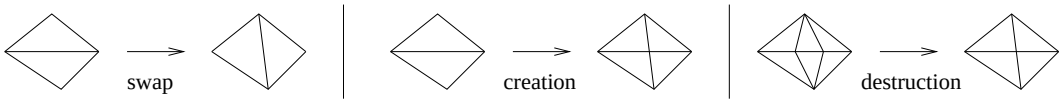

Figure 8: Eulerian topological transformations on triangulations.

This approach is relatively costly since it require the comparison of the distance between each vertex pair. An octree structure is used to optimize the computations.

6 Simplex Meshes

Simplex meshes where designed to represent deformable contours and surfaces [39]. They provide a simple and powerful framework to define regularization constraints. Simplex meshes allow the representation of surfaces with any topology, closed or opened. We use local forces with an explicit numerical scheme to efficiently compute model deformations. Simplex meshes are used for anatomical structures modelling and 3D and 4D medical image segmentation.

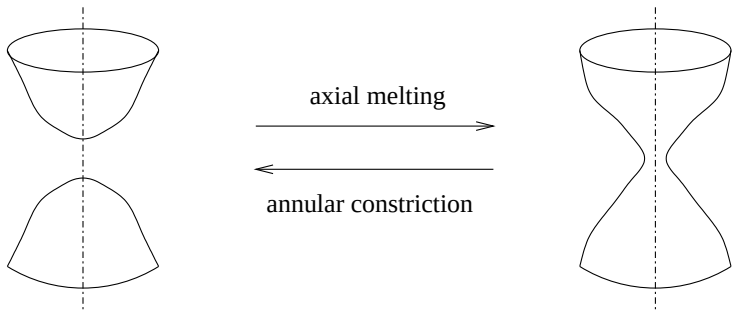

Figure 9: Non Eulerian topological transformations on triangulations.

6.1 Definition

A simplex mesh $\mathcal{M}$ is defined by a set of vertices $\{\mathbf{p}_i\}_i$ and a specific connectivity function. Each vertex of a k-simplex mesh is connected to exactly $k+1$ neighbors. A 1-simplex mesh is a simple deformable contour composed of a polygonal line. A 2-simplex mesh is a deformable surface for which each vertex is connected to exactly three neighbors as shown in figure 10.

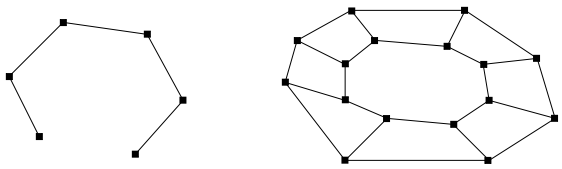

Figure 10: Left: opened 1-simplex mesh. Right: closed 2-simplex mesh.

The connectivity functions define mesh edges. There is at most one edge between two distinct vertices. A mesh face is composed by a an edge sequence over the mesh such as no edge split a face into two smallest faces. A border of a k-simplex mesh is a $(k-1)$-simplex mesh that creates a hole in the mesh. A border of a 1-simplex mesh is an edge labelled missing. Each border of a 2-simplex mesh is a face labelled missing (see figure 11). A detailed description of simplex meshes is given in [42].

6.1.1 Geometry

Due to its specific connectivity, a 2-simplex mesh is topologically dual to a triangulation. Figure 12 (left) shows a 2-simplex mesh example (solid line) and a dual triangulation (dashed line).

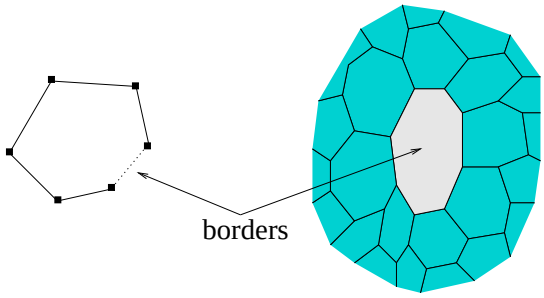

Figure 11: 1- and 2-simplex mesh with a border.

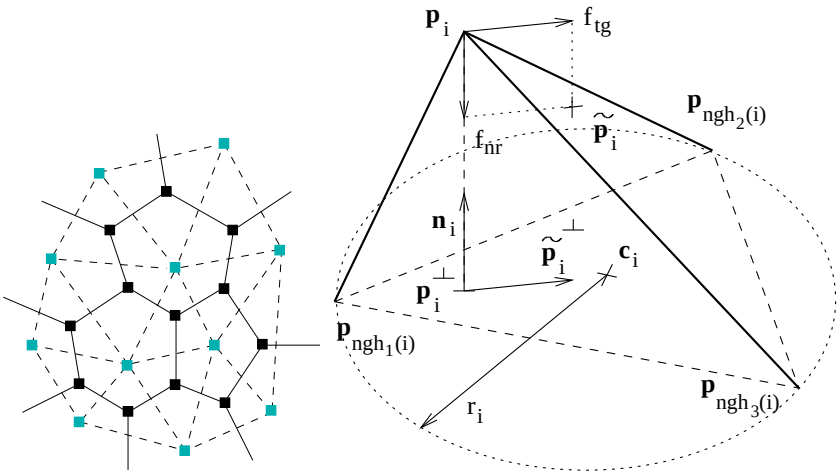

Figure 12: Right: 2-simplex mesh duality with triangulations. Left: Geometry and definition of the regularizing force.

The main features of simplex meshes lies in their simple geometric description. Figure 12 (right) shows a vertex $\mathbf{p}_i$ of a 2-simplex mesh and its three neighbors, $\mathbf{p}_{\mathrm{ngh}_j(i)}$. Let $\mathcal{P}_i$ be the plane defined by $\mathbf{p}_i$'s three neighbors. We denote $\mathbf{p}_i^\perp$ the projection of $\mathbf{p}_i$ on $\mathcal{P}_i$ and $\mathbf{n}_i$ the unit normal vector of $\mathcal{P}_i$. We introduce the circumscribed circle to triangle $(\mathbf{p}_{\mathrm{ngh}_1(i)}, \mathbf{p}_{\mathrm{ngh}_2(i)}, \mathbf{p}_{\mathrm{ngh}_3(i)})$ with center $\mathbf{c}_i$ and radius r_i, and the circumscribed sphere to vertices $(\mathbf{p}_i, \mathbf{p}_{\mathrm{ngh}_1(i)}, \mathbf{p}_{\mathrm{ngh}_2(i)}, \mathbf{p}_{\mathrm{ngh}_3(i)})$ with center $\mathbf{o}_i$ and radius R_i.

We call *metric parameters* the center of mass coordinates ε_i^1, ε_i^2, and $\varepsilon_i^3 = 1 - \varepsilon_i^1 - \varepsilon_i^2$ of $\mathbf{p}_i^\perp$ with respect to $\mathbf{p}_i$'s neighbors. They control the relative position of $\mathbf{p}_i^\perp$ in $\mathcal{P}_i$. The *simplex angle* $\varphi_i \in [-\pi, \pi]$ is defined by :

$$\begin{cases} \sin(\varphi_i) = \frac{r_i}{R_i}\mathrm{sign}((\mathbf{p}_{\mathrm{ngh}_1(i)} - \mathbf{p}_i) \cdot \mathbf{n}_i) \\ \cos(\varphi_i) = \frac{\|\mathbf{c}_i - \mathbf{o}_i\|}{R_i}\mathrm{sign}((\mathbf{c}_i - \mathbf{o}_i) \cdot \mathbf{n}_i). \end{cases}$$

It controls the elevation of vertex $\mathbf{p}_i$ above $\mathcal{P}_i$. We define a vertex discrete mean curvature as $H_i = \frac{1}{R_i} = \frac{\sin(\varphi_i)}{r_i}$. Under some assumptions it can be shown [38] that the discrete curvature of a mesh whose vertices lie on a continuous and sufficiently differentiable surface converges towards the surface mean curvature.

The vertex position $\mathbf{p}_i$ is uniquely defined by its three neighbors, its metric parameters and its simplex angle :

$$\mathbf{p}_i = \left(\sum_{j=1}^{3} \varepsilon_i^j \mathbf{p}_{\mathrm{ngh}_j(i)} \right) + H(\mathbf{p}_{\mathrm{ngh}_j(i)}, \varepsilon_i^j, \varphi_i)\mathbf{n}_i.$$

where:

$$H = \frac{(r_i^2 - d_i^2)tan(\varphi_i)}{\epsilon\sqrt{r_i^2 + (r_i^2 - d_i^2)tan(\varphi_i)^2} + r_i},$$

$$\epsilon = \begin{cases} 1 \text{ if } |\varphi_i| < \frac{\pi}{2} \\ -1 \text{ if } |\varphi_i| > \frac{\pi}{2} \end{cases}, \quad d_i = \|\mathbf{p}_i^\perp \mathbf{c}_i\|.$$

It can be shown [38] that a simplex mesh shape is defined up to a similarity transformation by the set of its metric parameters and simplex angles $\{\varepsilon_i^1, \varepsilon_i^2, \varphi_i\}_i$.

6.1.2 Law of motion

Each vertex is submitted to an internal force (f_{int}) and an external force (f_{ext}). We use a Newtonian law of motion (see equation 20). The explicit scheme discretization leads to equation 21 which is rewritten:

$$\mathbf{p}_i^{t+1} = \mathbf{p}_i^t + (1 - \gamma)(\mathbf{p}_i^t - \mathbf{p}_i^{t-1}) + \alpha_i f_{\mathrm{int}}(\mathbf{p}_i^t) + \beta_i f_{\mathrm{ext}}(\mathbf{p}_i^t),$$

where α_i and β_i are force weights including the vertex mass and the time step, and γ is a damping factor. The stability of this scheme is guaranteed if α_i, β_i, and γ range inside given bounds.

6.2 Regularization

Shape regularization is a key issue in deformable modelling. Regularization of the deformation process can be controlled at different scales.

6.2.1 Local regularization

Due to its discrete nature, the regularization of a simplex mesh is not based on the evaluation of surface partial derivatives but on the relative position of a vertex with respect to its neighbors, i.e. in terms of metric parameters and simplex angles. More precisely, each vertex $\mathbf{p}_i$ is attracted towards a point $\tilde{\mathbf{p}}_i$ (see figure 12, right) on a smoother mesh. Let $\tilde{\varepsilon}_i^j$, $\tilde{\varphi}_i$ and $\tilde{\mathbf{p}}_i^{\perp}$ denote the metric parameters, the simplex angle and the projection of $\tilde{\mathbf{p}}_i$ on $\mathcal{P}_i$ respectively. The internal force can be decomposed as the sum of a tangential and normal component :

$$\begin{aligned} f_{\text{int}}(\mathbf{p}_i) &= f_{\text{tg}}(\mathbf{p}_i) + f_{\text{nr}}(\mathbf{p}_i) \\ &= (\tilde{\mathbf{p}}_i^{\perp} - \mathbf{p}_i^{\perp}) + (H(\mathbf{p}_{\text{ngh}_j(i)}, \varepsilon_i^j, \varphi_i) - H(\mathbf{p}_{\text{ngh}_j(i)}, \tilde{\varepsilon}_i^j, \tilde{\varphi}_i))\mathbf{n}_i. \end{aligned}$$

The tangential component of the internal force controls the vertex spacing over the surface. To ensure uniformly spread vertices, metric parameters are all set equal : $\tilde{\varepsilon}_i^1 = \tilde{\varepsilon}_i^2 = \tilde{\varepsilon}_i^3 = \frac{1}{3}$. The normal component constrains the mean curvature of the surface through the simplex angle. The definition of $\tilde{\varphi}_i$ depends on the level of geometric regularity that should be enforced.

Let $\mathcal{N}_s(i)$ be the set of all vertices connected to vertex number i by an edges path whose length is less than s edges. The scale parameter, s, defines the neighborhood size over which the mesh is regularized and plays an important role. For medical image segmentation we usually consider :

Mean curvature continuity

. To ensure that the vertex discrete mean curvature converges towards the weighted average mean curvature of its neighborhood, we set

$$\tilde{\varphi}_i = \arcsin\left(r_i \sum_{j \in \mathcal{N}_s(i)} e_{ij} \frac{\sin(\varphi_j)}{r_j}\right) \text{ with } \sum_{j \in \mathcal{N}_s(i)} e_{ij} = 1.$$

This smoothness constraint is used when no anatomical shape information is available. The shape is supposed to be smooth.

Shape constraint

[83]. Let $\{\varphi_i^\circ\}_i$ be the set of simplex angles defining the reference shape of an anatomical structure. Setting $\tilde{\varphi}_i = \varphi_i^\circ$ constrains the surface to converge towards the reference shape in the absence of external forces. This constraint should be used when prior shape information is available.

6.2.2 From local to global regularization

Many approaches have been proposed to combine a global deformation procedure limiting the model shape variations and a local deformation component to match small shape variations [117, 105, 113]. In [84], we introduce a computationally efficient scheme combining a global transformation estimation (equation 24) and local deformations 20). The global transformation T is converted to a global and regular force field : $f_{\text{global}}(\mathbf{p}_i) = T(\mathbf{p}_i) - \mathbf{p}_i$. The surface is submitted to internal, external and global forces. A *locality* parameter λ weighs the influence of each force:

$$\begin{aligned} \mathbf{p}_i^{t+1} &= \mathbf{p}_i^t + (1-\gamma)(\mathbf{p}_i^t - \mathbf{p}_i^{t-1}) + \\ &\quad \lambda(\alpha_i f_{\text{int}}(\mathbf{p}_i^t) + \beta_i f_{\text{ext}}(\mathbf{p}_i^t)) + (1-\lambda) f_{\text{global}}(\mathbf{p}_i^t). \end{aligned} \tag{26}$$

From equation 26 it can be seen that when $\lambda = 0$, the deformations are purely global, and when $\lambda = 1$, the deformations are only local. Any other value of λ leads to an intermediate behavior. This approach is efficient since external forces are computed only once for local and global forces estimation. The global transformation is estimated using a closed form solution.

This method leads to a coarse to fine approach similar to the Blake and Zisserman's Graduated Non-Convexity algorithm [15]. When λ is 0, the deformation problem is rather convex and the convergence is robust. As λ increases, several local minima appear until the original (local) equation is used. At each iteration, the current deformation optimum is used as an initialization for the model.

6.3 Initialization

The initialization of deformable models has often been considered as the main limitation of that approach. In some medical applications, it is possible to estimate a prior shape and orientation of the anatomical structure of interest, knowing the patient orientation inside the acquisition device. Global transformations may then be used to compensate for the small orientation and scale variations between a rough initialization and the patient image.

Nevertheless, it may happen that the initial position and/or shape of the anatomical structure is not known. We then propose an automatic procedure to build a simplex mesh directly from the volumetric image [41]. First, the image needs to be binarized, using thresholding and mathematical morphology operators, to reveal a rough approximation of the structure of interest. A two stages algorithm is used to initialize the mesh from the binary image. Figure 13 (left) shows an example on an MR Image of the hand.

- A surface tracking algorithm first extracts a surface from the object boundary voxels (center). The resulting surface needs to be a discrete 2D manifold.
- A sampling algorithm preserving the surface topology then produces a mesh at a given resolution. It propagates regions from randomly selected seeds over the surface. By splitting regions when needed, a simplex mesh is created. The result is obtained by

adapting the mesh topology to the desired level of detail and deforming the mesh in the original image (right).

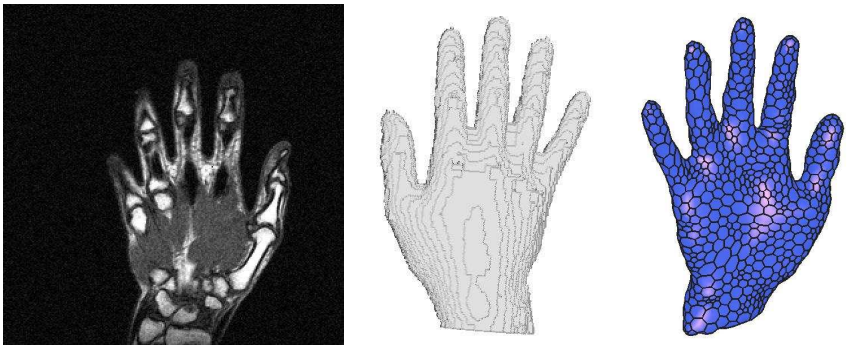

Figure 13: Simplex mesh automatic initialization in an MR Image of a hand.

6.4 Topology

Simplex meshes can represent surfaces with any topology. The topology of a surface simplex mesh can evolve according to 4 elementary operations that preserve the mesh regularity [42]. We designed algorithms to perform both mesh resampling (Eulerian operations) and changes in the surface genus (non-Eulerian operations).

6.4.1 Surface adaptation and refinement

Mesh sampling is key to control the model level of detail. Simplex mesh resampling is done using TO_1 and TO_2 topology operations illustrated in figure 14. TO_1 removes a vertex (respectively an edge) from a 1-simplex mesh (respectively a 2-simplex mesh). TO_2 is the dual operation that adds a vertex (respectively an edge) to a simplex mesh.

The mesh should also have good geometric and topological properties to give a usable geometric representation of a surface. We use two criteria to measure a simplex mesh quality:

- The *topological quality* is defined as the regularity of the number of edges in every mesh face. Figure 15 shows two different meshes representing the same sphere. The left mesh has a great topological quality while the center mesh has very irregular faces.
- The *geometric quality* measures the regularity of the vertices spacing.

Topological optimization

By composing a TO_2 and a TO_1 topological operation, one obtain the topology swap operation illustrated in figure 16. A swap operation conserve the number of mesh vertices,

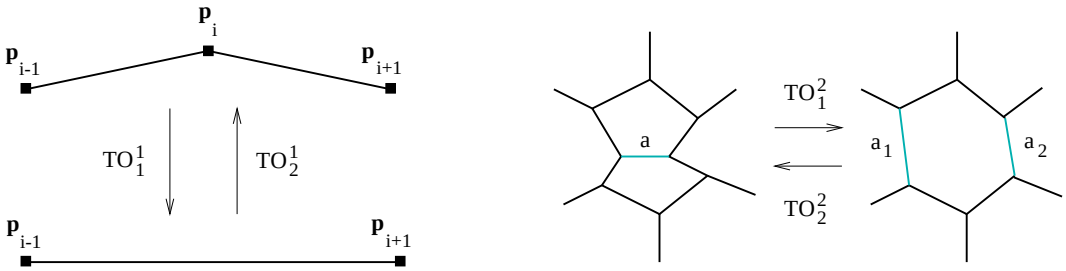

Figure 14: Eulerian topological operations on 1- (left) and 2- (right) simplex meshes.

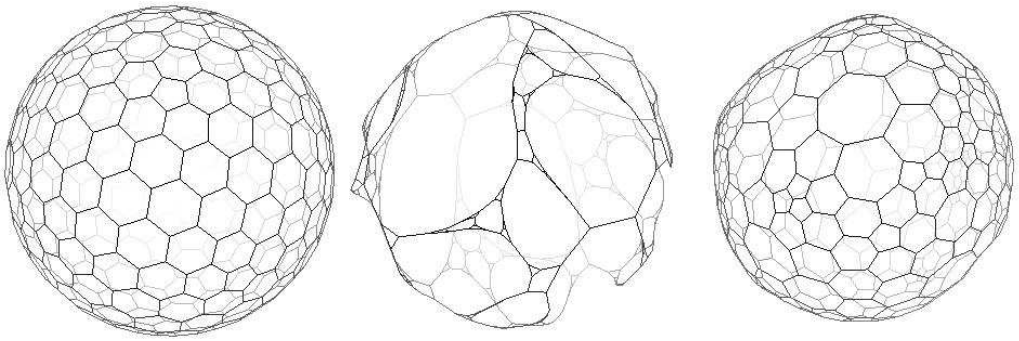

Figure 15: Sphere simplex meshes. Left: good topological quality. Center: bad topological quality. Right: topological quality restored from center mesh using swap operations.

edges and faces. However it changes the number of edges of each face (faces f_1 and f_2 loose an edge while faces f_3 and f_4 win an edge). It is used to optimize a mesh topological quality.

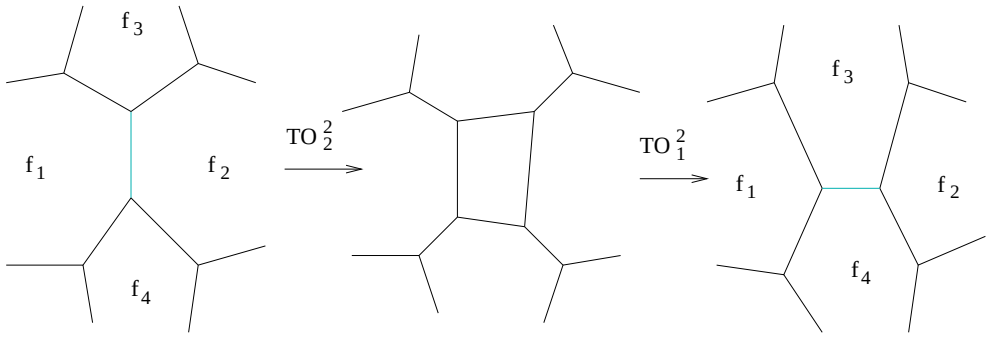

Figure 16: Topological swapping operation.

Let $N(f_i)$ denote the number of edges of face f_i. For each mesh edge e_j, the topological criterion $T(e_j) = N(f_1) + N(f_2) - N(f_3) - N(f_4)$ is computed. If $T(e_j) > 2$ then the edge is swapped. By repeating the procedure until no edge can be swapped, the mesh topological quality is restored as can be seen on figure 15. The center mesh represent a poor quality spherical surface. The right mesh is obtained from the center mesh by applying swap operations.

Global refinement

In general, it is desired to minimize the size of the geometric representation (a compact model) with sufficient details to represent the finest image structures. A multi-scale approach is often used to reconstruct data [64]. A rough mesh is first deformed and the surface is iteratively refined for faster convergence of the first deformation stages. We use a simplex mesh refinement/decimation algorithm obtained from a classical triangulation refinement/decimation method by duality. Figure 17 shows a coarse mesh (dark line) and the refined mesh (gray line). The left columns illustrate the refinement algorithm on a surface simplex mesh while the right column represent the algorithm on a dual triangulation $\mathcal{T}$.

The $\mathcal{T}$ triangulation is refined to $\mathcal{T}'$ in the following in three steps:

- A vertex of $\mathcal{T}'$ is created for each $\mathcal{T}$ vertex and each $\mathcal{T}$ triangle.
- An edge of $\mathcal{T}'$ connects each vertex of $\mathcal{T}$ with the center of each neighbor triangle.
- An edge of $\mathcal{T}'$ connects every centers of neighbor triangles of $\mathcal{T}$.

Let $n_v(\mathcal{T})$, $n_e(\mathcal{T})$, $n_f(\mathcal{T})$ and $n_g(\mathcal{T})$ denote the number of triangulation vertices, edges, faces of $\mathcal{T}$ and $\mathcal{T}$ genus respectively. $\mathcal{T}'$ owns one vertex for each vertex and each triangle of $\mathcal{T}$ thus:

$$n_v(\mathcal{T}') = n_v(\mathcal{T}) + n_f(\mathcal{T}).$$

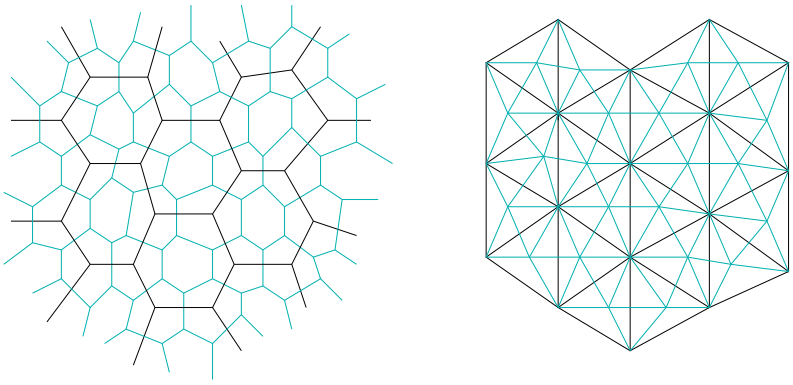

Figure 17: Left: 2-simplex mesh refinement. Right: dual operation on a dual triangulation.

From the Euler relation ($2n_v(\mathcal{T}) = n_f(\mathcal{T}) + 4(1 - n_g(\mathcal{T}))$), the number of vertices of $\mathcal{T}'$ is approximatively three times bigger than $n_v(\mathcal{T})$:

$$n_v(\mathcal{T}') = 3n_v(\mathcal{T}) - 4(1 - n_g(\mathcal{T})).$$

Let $\mathcal{M}$ and $\mathcal{M}'$ denote two simplex meshes dual to $\mathcal{T}$ and $\mathcal{T}'$ respectively. By duality, $n_f(\mathcal{M}) = n_v(\mathcal{T})$, $n_v(\mathcal{M}) = n_f(\mathcal{T})$ and $n_e(\mathcal{M}) = n_e(\mathcal{T})$. Consequently, the refinement operation approximatively multiplies the number of faces by 3:

$$n_f(\mathcal{M}') = 3n_f(\mathcal{M}) - 4(1 - n_g(\mathcal{M})).$$

The refinement operation decomposes in a succession of TO_2 and swap operations for each mesh face as illustrated in figure 18. Let us consider one mesh face (a). Two TO_2 operations are applied on two edges of each vertex around the face (b) and at a topological distance 1 of the face (c). A new triangular face is thus created for each vertex. Swap operations are applied on edges indicated by arrows in figure (c) and (d). This results in the refined face (f). The whole mesh is iteratively refined while avoiding to consider twice the same vertex.

The refinement operation and the dual decimation operation keep the topological quality of the mesh constant. The geometrical quality is preserved by creating new vertices such that they remain evenly spaced.

Figure 19 and 20 show two examples of medical image segmentation using a coarse to fine approach. Figure 19 shows a simplex mesh sphere composed of 500 vertices embedded in a 3D angioscanner of the heart[1]. The model is deformed but due to its coarse sampling it cannot represent the vessels thin details. It is refined twice to produce a 1500 vertices and a 4500 vertices mesh. The final model gives a detail representation of the endocardium and

[1] courtesy of the Mayo Clinic, `http://www.mayo.edu`

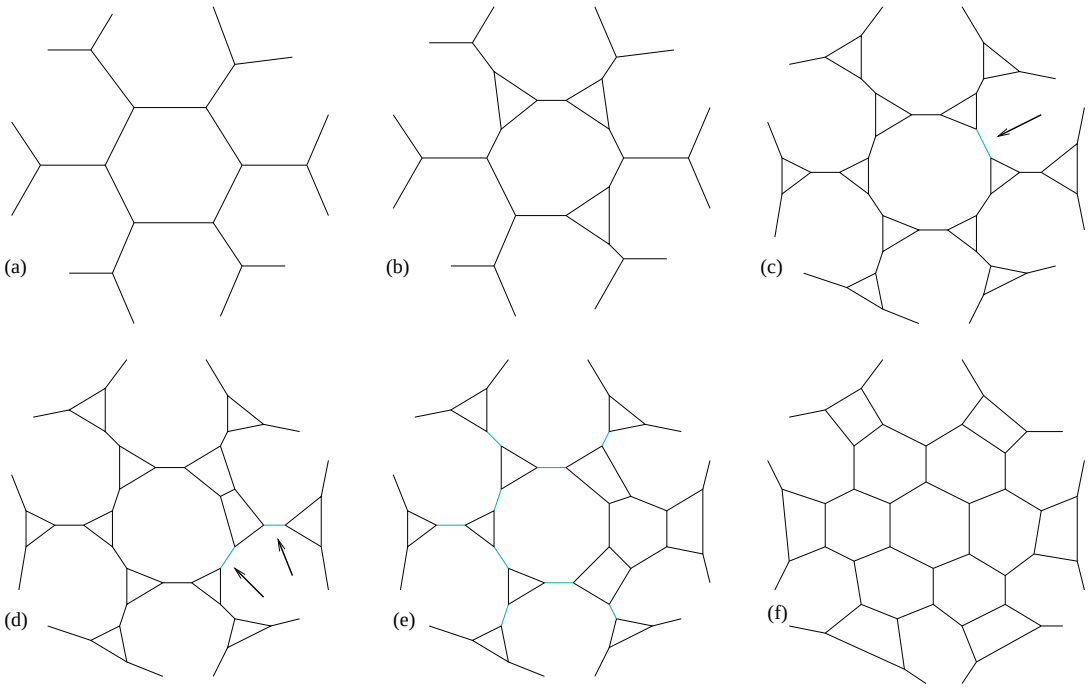

Figure 18: Refinement of a 2-simplex mesh face.

the main blood vessels but it shows a poor geometrical quality as vertices concentrate on the center due to the model initialization.

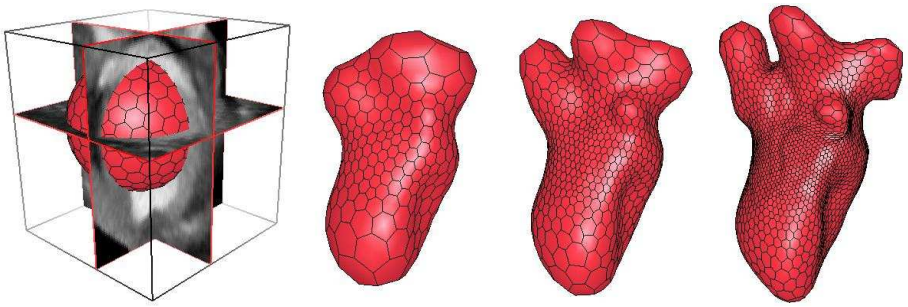

Figure 19: Endocardium segmentation by a coarse to fine approach.

Figure 20 shows the segmentation of the cortex from an MR image[2]. The initial 500 vertices mesh is refined twice as previously. The initial model deformations guarantee that new vertices are created close to their equilibrium position.

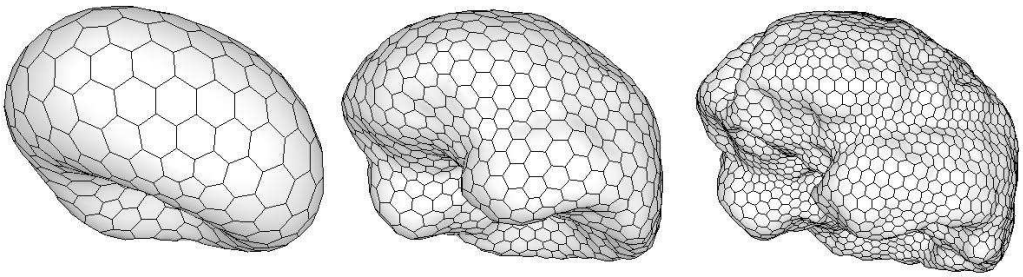

Figure 20: Cortex segmentation by a coarse to fine approach.

Geometrical optimization

It may be desired to force every faces of a mesh to converge towards equal surface polygons. This would avoid to degrade the mesh geometric quality such as illustrated in figure 19. On the contrary, it may be desired to concentrate vertices in areas with a high curvature while relatively flat areas do not require much vertices. This optimizes the number of vertices needed to represent a given shape but it is likely to make the model sensible to noise and outliers. The area and curvature optimization constraints are opposite and depend

[2]courtesy of the Brigham and Women's Hospital, `http://www.partners.org/bwh/home.html`

on the nature of the data and the desired result. Figure 21, left, shows the endocardium surface reconstructed in figure 19 after it has been optimized for area. The right figure shows a surface mesh optimized for curvature.

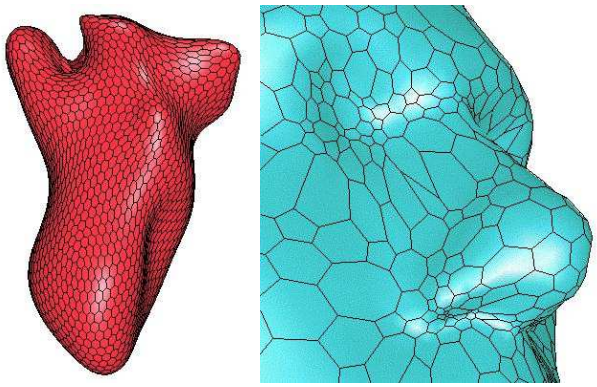

Figure 21: Left: endocardium with evenly spaced vertices. Right: face model with vertices concentrated in high curvature areas.

The simplex mesh metric parameters control the vertices relative position. To enforce a regular spacing of vertices, we compute metric parameters such as the area of every faces equalises. Each vertex i of a surface simplex mesh is surrounded by 3 faces $f_{FP_j(i)}, j \in \{1,2,3\}$. Let $A(f_{FP_j(i)})$ denote the area of face $f_{FP_j(i)}$ and $\bar{A}_i = \sum_j A(f_{FP_j(i)})/3$ the mean surface of neighbor faces of i. By setting:

$$\forall j, \quad \tilde{\varepsilon}_i^j = \frac{1}{3} + \kappa \frac{A_i(j)}{\bar{A}_i},$$

where κ is a constant, the tangential component of the internal force make the $\mathbf{p}_i^{\perp}$ projection point converge towards the point whose center of mass coordinates are $\tilde{\varepsilon}_i^j$ in $\mathbf{p}_i$ neighbors coordinate frame. Each vertex evolves until the its neighbor face's area equalise and the neighbor constraints propagate this behavior all over the mesh.

The mesh geometrical quality is also sensitive to the local refinement or decimation of the mesh. It is possible to remove vertices in areas of high concentration or to add vertices in areas where faces have a too large surface. Two geometrical criterions are used to control vertices refinement and decimation:

- *Area criterion.* If a face area becomes to large (respectively too small) compared to the mean faces area, the face is refined (respectively decimated). Let κ be a constant, f_i a mesh face and $A(f_i)$ its area:

$$\begin{cases} \frac{f_i}{\bar{A}} < \frac{1}{\kappa} \Rightarrow \text{fusion of } f_i \text{ and the neighbor face with the smallest area} \\ \frac{f_i}{\bar{A}} > \kappa \Rightarrow \text{splitting of } f_i. \end{cases}$$

- *Elongation criterion.* The elongation parameter of a face depends on the respective length of its edges. Let $\{e_i\}$ be the set of face f edges. Then:

$$elongation(f) = \frac{max_i \|e_i\|}{median_i \|e_i\|}.$$

If $elongation(f)$ is greater than a constant ratio κ then the face is split in its narrowest area.

Figure 22 shows the effect of the metric parameter area equalisation constraint and the geometrical constraints. Top row shows the initial surface with a local force applied on a vertex (left) and the model deformation without any geometrical constraint. Bottom row, left, shows the mesh deformed with the metric parameter area equalisation constraint. All face keep the same area but some faces have a poor geometrical quality due to the surface elongation. Bottom row, right, shows the deformation of the surface controlled by the geometrical refinement criterion. The geometrical quality is better than in the previous case but the faces area are not as regular. Those two constraints are complementary and can be used simultaneously.

6.4.2 Topology changes

Non Eulerian primitive operations perform topology changes on simplex meshes [39]. In [44] we describe an algorithm based on discretization of space on a regular grid that automatically adapts a 1-simplex mesh topology to the image data. Figure 23 shows an example of the automatic topology change algorithm for the segmentation of a vertebra in an X-ray image. The closed contour deforms and split in two connected components to fit the vertebra topology.

7 Medical images segmentation

7.1 Image geometry

Most medical images are sampled along a regular grid in Euclidean space. However, some acquisition devices acquire volumetric images using different geometries. In particular, 3D ultrasound probes often acquire a set of 2D planes that may not be parallel to each other. Figure 24 shows a rotative ultrasound probe and the corresponding cylindrical sampling grid. In these images, the data density decreases as a function of the distance to the symmetry axis. In [85] we proposed an algorithm for filtering images with a non Euclidean geometry. This is especially useful to extract gradient information which is the main feature used for image segmentation.

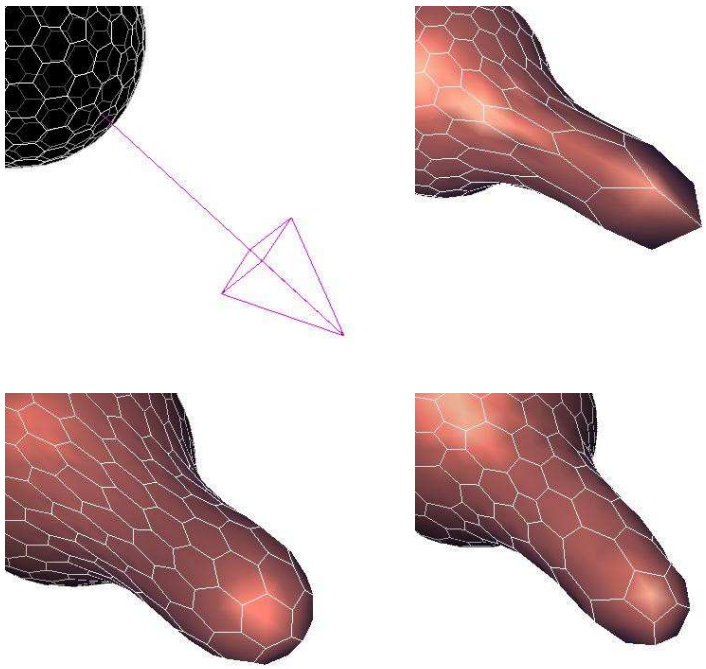

Figure 22: Deformable surface response to a local perturbation. From top to bottom, left to right: local force applied; deformation without geometrical constraint; metric parameters area equalisation; geometrical refinement criterion.

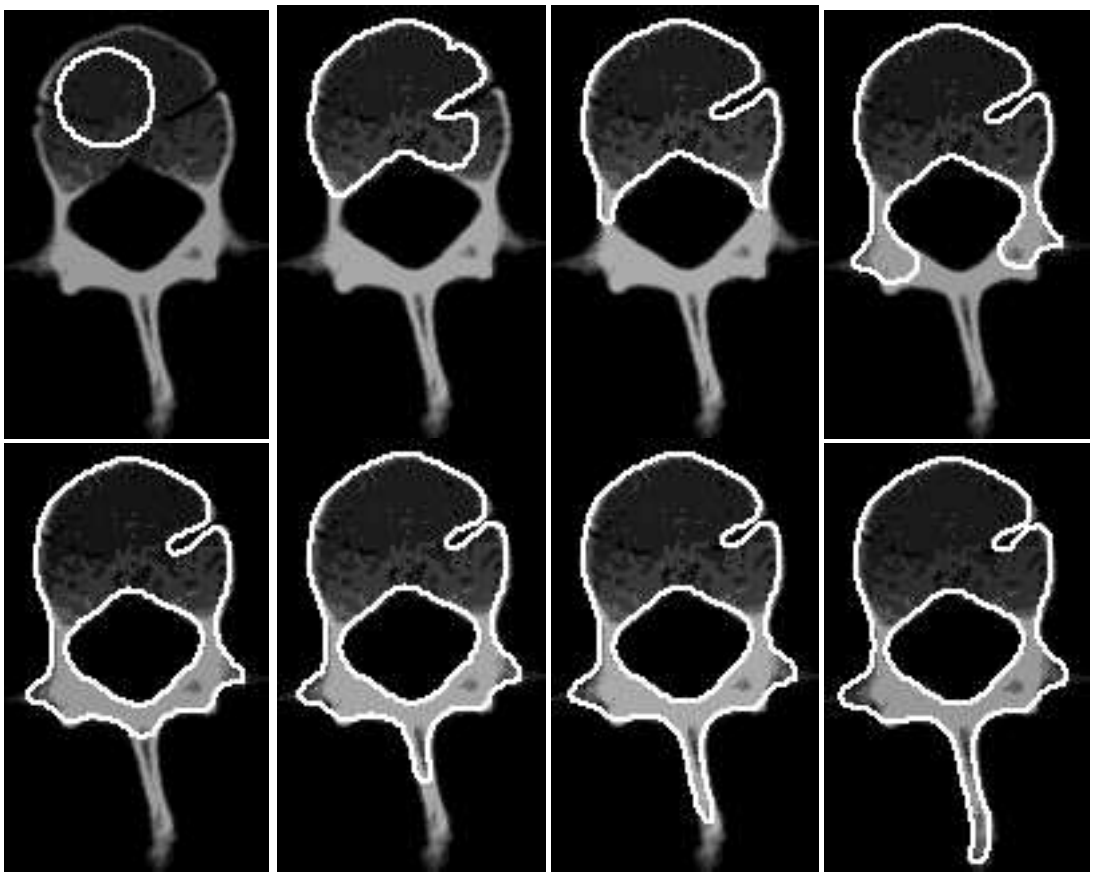

Figure 23: Segmentation of a vertebra with change in topology of the original contour.

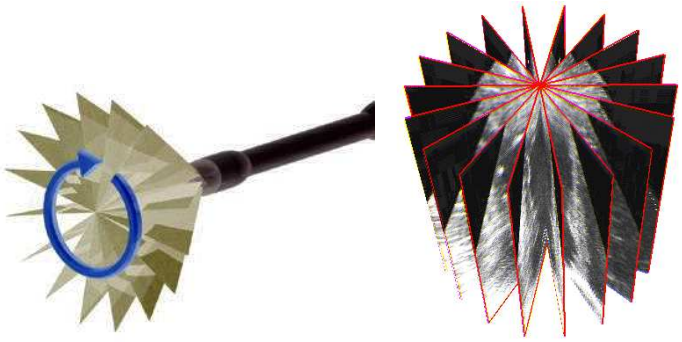

Figure 24: A rotative ultrasound probe and the resulting cylindrical geometry image. This image was acquired at the CHU Brabois by Dr Lethor for the Échocard3D project.

7.2 External forces

Although the use of global deformations improves the segmentation robustness, high segmentation accuracy can be achieved by introducing some specific knowledge about the structures of interest.

External forces are computed from an image I to push a deformable model towards anatomical structure boundaries. In their seminal paper [60], Kass *et al* proposed to use the gradient of the image gradient's norm as the external force : $f_{\text{ext}} = \nabla\|\nabla I\|$. The potential field associated with this force tends to be fairly narrow around image edges. Moreover, the model evolution might become unstable. Cohen [27] proposed a normalization to avoid instabilities and adds a balloon force that causes the model inflation. The use of a function of the distance from a model vertex to the edge points also avoids instabilities. Thus forces computation often requires the extraction of main edges. A distance map is then computed from the edges or a closest edge point is determined for each model vertex.

7.2.1 Gradient based forces

We use different external forces depending on the image quality and specificity. For most CT and MR images, the contrast is high enough to extract boundaries using a gradient operator. The model then tracks close boundaries at each pixel by scanning image voxels along the normal direction at each vertex. The scan-line algorithm is illustrated for Euclidean and cylindrical images in figure 25. Restricting the boundary search along the model normals reduces the scanning time without limiting the deformation capabilities.

If the scan-line from a vertex intersects a voxel of high gradient norm, the external force is computed as the displacement vector from the vertex position to the boundary voxel center. To make the boundary detection more robust, several criteria can be taken into account :

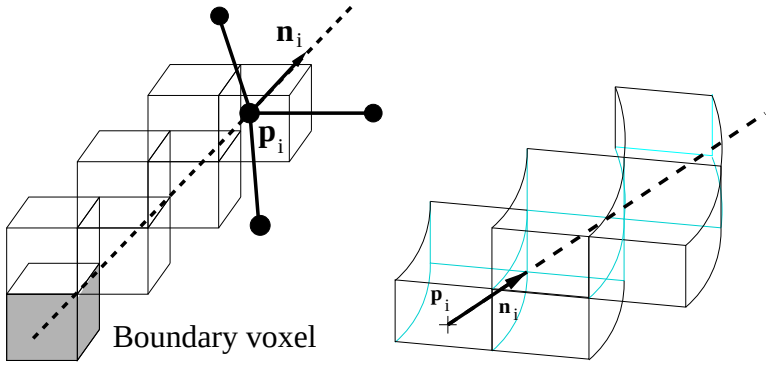

Figure 25: Scan-line algorithm for different image geometries.

- *Range.* The scan-line algorithm scans voxels within a limited range to avoid reaching outliers when gradient voxels are lacking. The range parameter is either dependent on the image size or of the average structure size when it is known.
- *Contrast.* In fairly well contrasted images, a lower gradient intensity threshold can be set to ignore voxels whose gradient's norm is too low (weak contours or noise).
- *Direction.* If the segmented structure is brighter or darker than the image background, the gradient vector is always pointing outside or inside that structure. A gradient voxel found on a vertex normal line is rejected if its direction does not roughly match the normal direction.
- *Intensity.* Finally, if the structures of interest intensity range is known, a gradient voxel can be rejected when the corresponding intensity voxel does not fall within the expected range.

7.2.2 Region based forces

In many cases, the gradient information may be lacking or the noise level may be too high to detect proper boundaries using a gradient operator. Moreover, gradient information is not necessarily relevant in some image acquisition systems, e.g. thick lines appear in ultrasound images along organ interfaces. An information on the gray-level distribution is then more appropriate.

We have developed an intensity region external force [85]. At each mesh vertex, an intensity profile is acquired from the image in the vertex normal direction. The profile is then smoothed using anisotropic diffusion to reduce noise without altering intensity discontinuities. When an homogeneous region (i.e. a profile segment with its voxels belonging to a given intensity range) having a minimum length is found, we search for voxels of high

gradient in the vicinity of the region's extremities. These voxels correspond to the region boundary and are used to compute the external force.

Figure 26 shows the heart left ventricle segmentation from a time series of very noisy 3D ultrasound images with weak contours based on region-defined external forces. First row shows a surface model initialized in the first image of the time sequence. The second row shows the recovered left ventricle shape at 4 instants of the cardiac cycle. The three last rows show three of the image planes at 4 different instants with the deformed models intersection superimposed. The time sequence corresponds to the heart contraction and the left ventricle volume variation curve computed from the model is displayed in figure's first row. Model based quantitative measurements are useful to determine relevant parameters in terms of pathology diagnosis such as ejection fraction.

8 Conclusion

Deformable surfaces provide a powerful tool for data modelling and image segmentation. They produce a geometric representation of complex structures suited to visualization, quantitative measurements and simulation. Different surface representations lead to various modeling capabilities. The representation controls the number of degrees of freedom let to the surface and has an effect on the possible deformations. The kind of regularizing constraints also depends on the surface intrinsic parameters.

Deformable surface shape can evolve to represent complex structures without any restriction of topology. Although deformable models may be trapped by local minima of the energy functional, their robustness is increased in many applications by the introduction of prior information and constraints in the deformation process.

The application field of image segmentation based on deformable models can be extended. For instance, the segmentation of time series of medical images is of high importance for many medical procedures including cardiology. We are developing time-dependent deformable models in order to obtain time-correlated image segmentation. Also, the reconstruction of vascular trees based on deformable models is challenging because of their complex geometry and tubular symmetry.

Acknowledgements

We are grateful to Michel Audette for comments and proof reading.

References

[1] R. Acharya, R. Wasserman, J. Sevens, and C. Hinojosa. Biomedical Imaging Modalities: a Tutorial. *Computerized Medical Imaging and Graphics*, 19(1):3–25, 1995.

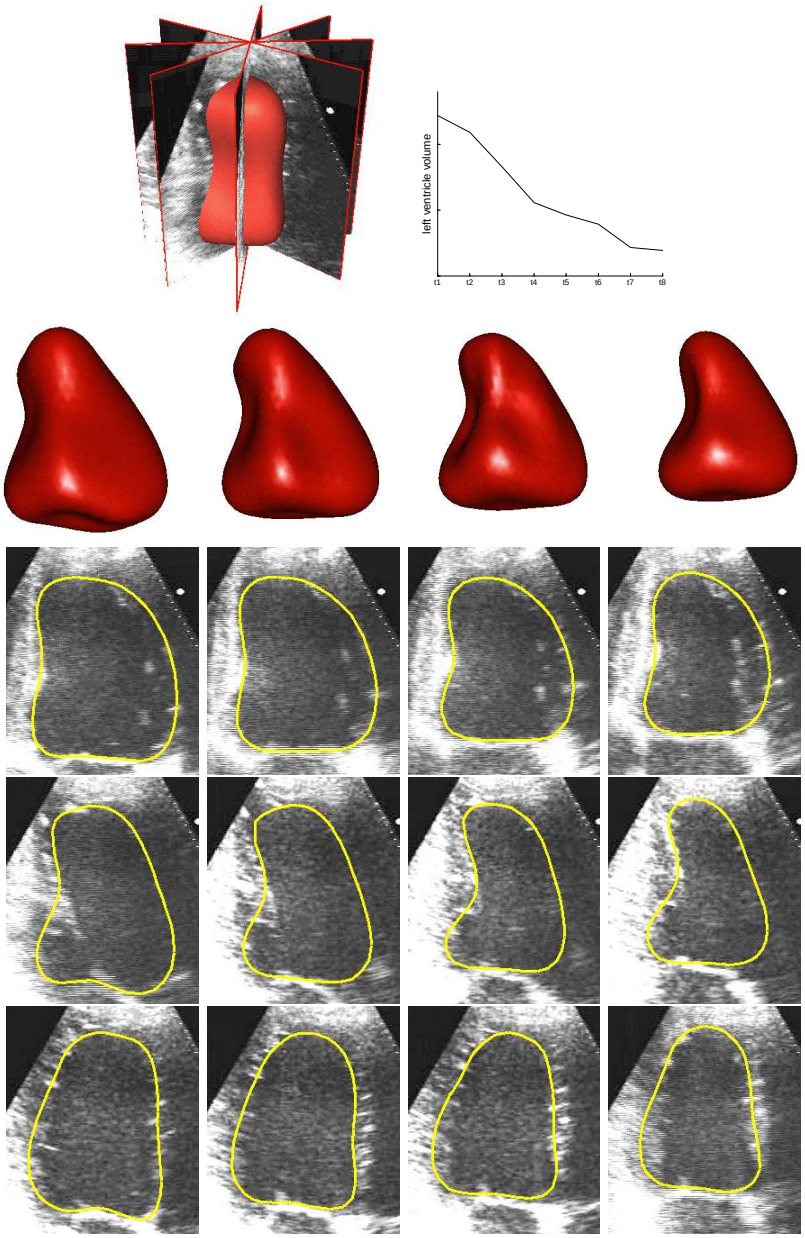

Figure 26: Segmentation of a 3D cardiac ultrasound images sequence.

[2] A. Aguado, M. Nixon, and M. Montiel. Parameterizing Arbitrary Shapes via Fourier Desciptors for Evidence-Gathering Extraction. *Computer Vision and Image Understanding*, 69(2):202–221, Feb. 1998.

[3] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[4] M. Audette, F. Ferrie, and T. Peters. An Algorithmic Overview of Surface Registration Techniques for Medical Imaging. *to appear in Medical Image Analysis*, 4, 2000.

[5] N. Ayache, P. Cinquin, I. Cohen, L. Cohen, F. Leitner, and O. Monga. Segmentation of Complex Three-Dimensional Medical Objects: A Challenge and a Requirement for Computer Assisted Surgery Planning and Performance. In B. Taylor, Lavallée and Mösges, editors, *Computer Integrated Surgery, Technology and Clinical Applications*, pages 59–74. MIT press, 1996.

[6] E. Bainville. Reconstruction d'objets tridimensionnels à partir de silhouettes. Master's thesis, École Normale Supérieure de Lyon, July 1992.

[7] B. Baldwin, D. Geiger, and R. Hummel. Resolution-Appropriate Shape Representation. In *International Conference on Computer Vision (ICCV'98)*, pages 460–465, Bombay, India, Jan. 1998.

[8] E. Bardinet. *Modèles déformables contraints : applications à l'imagerie cardiaque.* PhD thesis, Université Paris-IX Dauphine, France, 1995.

[9] E. Bardinet, D. Cohen, and N. Ayache. A Parametric Deformable Model to Fit Unstructured 3D Data. Technical Report 2617, INRIA, July 1995.

[10] E. Bardinet, L. Cohen, and N. Ayache. Tracking and motion analysis of the left ventricle with deformable superquadrics. *Medical Image Analysis*, 1(2):129–149, 1996.

[11] B. Bascle and R. Deriche. Region tracking through image sequences. Technical Report 2439, INRIA, Dec. 1994.

[12] P. Besl and N. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb. 1992.

[13] A. Blake, R. Curwen, and A. Zisserman. Affine-invariant contour tracking with automatic control of spatiotemporal scale. In *International Conference on Computer Vision (ICCV'93)*, pages 66–75, Berlin, Germany, May 1993.

[14] A. Blake and I. Michael. 3D position, attitude and shape input using video tracking of hands and lips. In *ACM Computer Graphics (SIGGRAPH'94)*, pages 185–192, Orlando, USA, 1994.

[15] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[16] F. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[17] L. Brown. A Survey of Image Registration Techniques. *ACM Computing Surveys*, 24(4):325–376, Dec. 1992.

[18] A. Bulpitt and N. Efford. An Efficient 3D Deformable Model with a Self Optimizing Topology. In *British Machine Vision Conference*, volume 1, pages 37–46, 1995.

[19] A. Caunce and C. Taylor. 3D Point Distribution Models of the Cortical Sulci. In *International Conference on Computer Vision (ICCV'98)*, pages 402–407, Bombay, India, Jan. 1998.

[20] B. Chalmond and S. Girard. Nonlinear Data Representation for Visual Learning. Technical Report 3550, INRIA, Nov. 1998.

[21] K.-W. Cheung, D.-Y. Yeung, and T. Chin. On deformable models for visual pattern recognition. Technical Report HKUST-CS98-8, Hong Kong University of Science and Technology, 1998.

[22] G. Christensen, R. Rabbitt, and M. Miller. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, 5(10):1435–1447, oct 1996.

[23] P. Clarysse, D. Friboulet, and I. Magnin. Tracking Geometrical Descriptors on 3-D Deformable Surfaces: Application to the Left-Ventricular Surface of the Heart. *IEEE Transactions on Medical Imaging*, 16(4):392–404, Aug. 1997.

[24] I. Cohen. *Modèles déformables 2D et 3D, application à la segmentation d'images médicales.* PhD thesis, Université Paris-IX Dauphine, France, June 1992.

[25] I. Cohen and L. Cohen. Hyperquadric model for 2D and 3D data fitting. In *12th International Conference on Pattern Recognition (ICPR'94)*, pages 403–405, Jerusalem, Israel, 1994.

[26] I. Cohen and L. Cohen. A Hybrid Hyperquadric Model For 2-D And 3-D Data Fitting. *Computer Vision and Image Understanding*, 63(3):527–541, May 1996.

[27] L. Cohen. On Active Contour Models and Balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, Mar. 1991.

[28] L. Cohen and I. Cohen. Finite element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, Nov. 1993.

[29] L. Cohen and A. Gorre. On the convexity of the active contour energy. In *proceedings of the GRETSI*, Juan-les-Pins, France, Sept. 1995.

[30] L. Cohen and R. Kimmel. Fast Marching The Global Minimum of Active Contours. In *International Conference on Image Processing (ICIP'96)*, volume A, pages 473–476, Lausanne, Switzerland, Sept. 1996.

[31] L. Cohen and R. Kimmel. Global Minimum For Active Contour Models: A Minimum Path Approach. In *International Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 666–673, San Fransisco, USA, June 1996.

[32] T. Cootes, A. Hill, C. Taylor, and J. Halsan. The Use of Active Shape Models for Locating Structures in Medical Images. In *International Conference on Information Processing in Medical Images (IPMI'93)*, volume 687 of *Lectures Notes in Computer Science*, pages 33–47, Flagstaff, USA, June 1993. Springer-Verlag.

[33] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models, their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, Jan. 1995.

[34] S. Coquillart. Extended free-form deformation: A sculpturing tool for 3D geometric modeling. *ACM Computer Graphics (SIGGRAPH'90)*, 24(4):187–196, Aug. 1990.

[35] C. Davatzikos and J. Prince. Adaptative active contour algorithms for extracting and mapping thick curves. In *International Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 524–529, New York, USA, 1993.

[36] J. Declerck, J. Feldmar, M. Goris, and F. Betting. Automatic registration and alignement on a template of cardiac stress and rest reoriented SPECT images. *IEEE Transactions on Medical Imaging*, 16(7):727–737, Dec. 1997.

[37] H. Delingette. Intrinsic stabilizers of planar curves. In *3rd European Conference on Computer Vision (ECCV'94)*, Stockholm, Sweden, June 1994.

[38] H. Delingette. *Modélisation, déformation et reconnaissance d'objets tridimensionnels à l'aide de maillages simplexes.* PhD thesis, École Centrale de Paris, France, July 1994.

[39] H. Delingette. Simplex Meshes: a General Representation for 3D Shape Reconstruction. Technical Report 2214, INRIA, Mar. 1994.

[40] H. Delingette. General Object Reconstruction based on Simplex Meshes. Technical Report 3111, INRIA, Feb. 1997.

[41] H. Delingette. Initialization of deformable models from 3D data. In *International Conference on Computer Vision (ICCV'98)*, pages 311–316, Bombay, India, Jan. 1998.

[42] H. Delingette. General Object Reconstruction based on Simplex Meshes. *International Journal of Computer Vision*, 32(2):111–146, 1999.

[43] H. Delingette and J. Montagnat. New algorithms for controlling active contours shape and topology. In *to appear in European Conference on Computer Vision (ECCV'00)*, Dublin, Ireland, June 2000.

[44] H. Delingette and J. Montagnat. Topology and shape constraints on parametric active contours. Technical Report 3880, INRIA, Jan. 2000.

[45] H. Delingette, G. Subsol, S. Cotin, and P. J. A craniofacial surgery testbed. Technical Report 2199, INRIA, Mar. 1992.

[46] M. Desbrun and M.-P. Gascuel. Smoothed-Particles: A new paradigm for animating highly defromable bodies. In *Eurographics Workshop on Animation and Simulation*, Poitier, France, Sept. 1996.

[47] C. Dorai and A. Jain. COSMOS - A Representation Scheme for 3D Free-Form Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, 1997.

[48] M. Eck and H. Hoppe. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In *Computer Graphics (SIGGRAPH'96)*, 1996.

[49] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. Technical Report 2220, INRIA, Mar. 1994.

[50] J. Feldmar and N. Ayache. Rigid, Affine and Locally Affine Registration of Free-Form Surfaces. *International Journal of Computer Vision*, 18(2):99–119, May 1996.

[51] D. Geiger, A. Gupta, A. Luiz, and J. Vlontzos. Dynamic Programming for Detecting, Tracking, and Matching Deformable Contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):294–302, 1995.

[52] S. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical Report TR-97-19, Mitsubishi Electric Research Laboratory, Nov. 1997.

[53] M. González Ballester, A. Zisserman, and M. Brady. Measurement of Brain Structures based on Statistical and Geometrical 3D Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'98)*, volume 1496 of *Lecture Notes in Computer Science*, pages 499–508, Cambridge, USA, Oct. 1998. Springer.

[54] A. Guéziec and N. Ayache. Smoothing and Matching of 3-D Space Curves. In *second European Conference on Computer Vision (ECCV'92)*, pages 620–629, May 1992.

[55] K. Han and D. Goldgof. Using hyperquadrics for shape recovery from range data. In *International Conference on Computer Vision (ICCV'93)*, pages 492–496, Berlin, Germany, May 1993.

[56] T. Heap and D. Hogg. Wormholes in Shape Space : Tracking through Discontinuous Changes in Shape. In *International Conference on Computer Vision (ICCV'98)*, pages 344–349, Bombay, India, Jan. 1998.

[57] A. Hill, A. Thornham, and C. Taylor. Model-Based Interpretation of 3D Medical Images. In *British Machine Vision Conference (BMVC'93)*, pages 339–348, Guildford, UK, Sept. 1993.

[58] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Computer Graphics (SIGGRAPH'94)*, pages 295–302, Orlando, USA, 1994.

[59] M. Kamber, R. Shinghal, D. L. Collins, G. S. Francis, and A. C. Evans. Model-based 3D segmentation of multiple sclerosis lesions in magnetic resonance brain images. *IEEE Transactions on Medical Imaging*, 14:442–453, Sept. 1995.

[60] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1:321–331, 1988.

[61] A. Kelemen, G. Székely, and G. Gerig. Three-dimensional Model-Based Segmentation of Brain MRI. In *Workshop on Biomedical Image Analysis (WBIA'98)*, pages 4–13, Santa-Barbara, USA, June 1998.

[62] C. Kervrann and F. Heitz. Apprentissage non supervisé et suivi de modèles déformables dans une séquence d'images. In *RFIA'96*, pages 559–568, Rennes, France, 1996.

[63] J.-O. Lachaud. *Extraction de surfaces à partir d'images tri-dimensionnelles : approche discrète et approche par modèle déformable.* PhD thesis, Université Joseph Fourier, Grenoble 1, France, 1998.

[64] J.-O. Lachaud and A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. *Medical Image Analysis*, 3(2):187–207, 1999.

[65] F. Leitner and P. Cinquin. Complex topology 3d objects segmentation. In *SPIE Conf. on Advances in Intelligent Robotics Systems*, volume 1609, Boston, Nov. 1991.

[66] F. Leitner and P. Cinquin. From snake and splines to snake-splines. In Springer-Verlag, editor, *Workshop on Geometric Reasoning for Perception and Action*, volume 708 of *Lecture Notes in Computer Science*, pages 264–281, Grenoble, France, 1993.

[67] S. Leitner, M. I., S. Lavallée, and P. Cinquin. Dynamic segmentation: finding the edge with spline snakes. In P.-J. Laurent, A. Le Méhauté, and L. Schumaker, editors, *Curves and Surfaces*, pages 279–284. Academic Press, 1991.

[68] J.-C. Lombardo. *Modélisation d'objets déformables avec un système de particules orientées.* PhD thesis, Université Joseph-Fourier, iMAGIS/IMAG, Grenoble, France, Jan. 1996.

[69] L. Lorigo, O. Faugeras, W. Grimson, R. Keriven, R. Kikinis, and C.-F. Westin. Co-dimension 2 Geodesic Active Contours for MRA Segmentation. In *International Conference on Information Processing in Medical Images (IPMI'99)*, volume 1613 of *Lectures Notes in Computer Science*, Visegrád, Hungary, June 1999. Springer.

[70] J. Lötjönen, I. Magnin, L. Reinhardt, J. Nenonen, and T. Katila. Automatic Reconstruction of 3D Geometry Using Projections and a Geometric Prior Model. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'99)*, volume 1679 of *Lectures Notes in Computer Science*, pages 192–201, Cambridge, UK, Sept. 1999. Springer.

[71] J. Lötjönen, P.-J. Reissman, I. Magnin, and T. Katila. Model extraction from magnetic resonance volume data using the deformable pyramid. *Medical Image Analysis*, 3(4):387–406, 1999.

[72] D. Luenberger. *Linear and Nonlinear Programming (2nd ed.)*. Addison-Wesley, 1984.

[73] R. Malladi and J. Sethian. A Real-Time Algorithm for Medical Shape Recovery. In *International Conference on Computer Vision (ICCV'98)*, pages 304–310, Bombay, India, Jan. 1998.

[74] R. Malladi, J. Sethian, and B. Vemuri. Shape Modeling with Front Propagation : A Level Set Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–174, 1995.

[75] J. Mallet. Discrete smooth interpolating in geometric modelling. *Computer Aided Design (CAD)*, 24(4):178–192, 1992.

[76] T. McInerney and D. Terzopoulos. A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In *International Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, volume 905 of *Lectures Notes in Computer Science*, pages 518–523. Springer-Verlag, Apr. 1995.

[77] T. McInerney and D. Terzopoulos. Medical Image Segmentation using Topologically Adaptable Snakes. In *International Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, volume 905 of *Lectures Notes in Computer Science*, pages 92–100. Springer-Verlag, Apr. 1995.

[78] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[79] T. McInerney and D. Terzopoulos. Medical Image Segmentation using Topologically Adaptable Snakes. In *Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed-MRCAS'97)*, volume 1205 of *Lectures Notes in Computer Science*, pages 92–100. Springer-Verlag, Mar. 1997.

[80] S. Menet, P. Saint-Marc, and G. Medioni. B-Snakes: implementation and application to stereo. In *Artificial Intelligence and Computer Vision*, pages 223–236. Elsevier Science, 1991.

[81] D. Metaxas and D. Terzopoulos. Constrained Deformable Superquadrics and nonrigid Motion Tracking. In *International Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 337–343, Maui, Hawai, June 1991.

[82] C. Monserrat Aranda, M. Juan Lizandra, M. Raya, V. Grau Colomer, and C. Knoll. Deformation Simulation Algorithms of Elastic Tissues in "Real-Time" Based in Elasticity Theory. In J. W. et al., editor, *Medicine Meets Virtual Reality (MMVR'99)*, pages 21–22. IOS Press, 1999.

[83] J. Montagnat and H. Delingette. Volumetric Medical Images Segmentation using Shape Constrained Deformable Models. In *Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed-MRCAS'97)*, volume 1205 of *Lectures Notes in Computer Science*, pages 13–22. Springer-Verlag, Mar. 1997.

[84] J. Montagnat and H. Delingette. Globally constrained deformable models for 3D object reconstruction. *Signal Processing*, 71(2):173–186, Dec. 1998.

[85] J. Montagnat, H. Delingette, and G. Malandain. Cylindrical Echocardiographic Images Segmentation based on 3D Deformable Models. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'99)*, volume 1679 of *Lectures Notes in Computer Science*, pages 168–175, Cambridge, UK, Sept. 1999. Springer.

[86] A. Moreau-Gaudry, P. Cinquin, and J.-P. Baguet. Active Model Based Carotid Ultrasonic Data Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'99)*, volume 1679 of *Lectures Notes in Computer Science*, pages 176–183, Cambridge, UK, Sept. 1999. Springer.

[87] G. Mozelle and F. Prêteux. La méthode des ondelettes mobiles. Applications aux modèles déformables et à la reconstruction de surfaces. In *RFIA'96*, pages 569–578, Rennes, France, 1996.

[88] C. Nastar and A. Ayache. Classification of Nonrigid Motion in 3D Images using Physics-Based Vibration Analysis. In *IEEE Workshop on Biomedical Image Analysis*, pages 61–69, Seattle, USA, 1994.

[89] C. Nastar and N. Ayache. Frequency-Based Nonrigid Motion Analysis: Application to Four Dimensional Medical Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1067–1079, 1996.

[90] C. Nastar, B. Moghaddam, and A. Pentland. Generalized Image Matching : Statistical Learning of Physically-Based Deformations. In *fourth European Conference on Computer Vision (ECCV'94)*, pages 589–598, 1996.

[91] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[92] X. Pennec. *L'incertitude dans les Problèmes de Reconnaissance et de Recalage. Application en Imagerie Médicale et Biologie Moléculaire.* PhD thesis, École Polytechnique, France, 1996.

[93] A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically Based Shape Modeling and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):715–729, July 1991.

[94] C. Poon, M. Braun, R. Fahrig, A. Ginige, and A. Dorrell. Segmentation of medical images using an active contour model incorporating region-based images features. In *Visualization in Biomedical Computing (VBC'94)*, volume 2359, pages 90–97, Rochester, USA, Oct. 1994. SPIE.

[95] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipies in C (2nd ed.)*. Cambridge University Press, 1992.

[96] P.-J. Reissman and I. Magnin. Modeling 3D deformable object with the active pyramid. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(7):1129–1139, 1997.

[97] A. Robert. Modelization of the Left Ventricle and its Deformations using Superquadrics and Hyperquadrics. In *Computer Assisted Radiology*, pages 189–194, Berlin, Germany, June 1995.

[98] A. Robert. *Étude de la forme et du mouvement du coeur à partir de données lacunaires.* PhD thesis, École Nationale Supérieure des Télécomunications, France, 1996.

[99] C. Ruff, A. Bhalerao, S. Hughes, T. D'Arcy, and D. Hawkes. The Estimation of Fetal Organ Volume using Statistical Shape Analysis. In *Computer Assisted Radiology*, pages 280–285, 1996.

[100] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *ACM Computer Graphics (SIGGRAPH'86)*, 20(4):151–160, Aug. 1986.

[101] J. Sethian. *Level Set Methods : Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Science.* Cambridge University Press, 1996.

[102] K. Siddiqi, A. Tannenbaum, and S. Zucker. Hyperbolic "smoothing" of shapes. In *International Conference on Computer Vision (ICCV'98)*, pages 215–221, Bombay, India, Jan. 1998.

[103] P. Sozou, T. Cootes, C. Taylor, and E. Di-Mauro. A Non-linear Generalisation of PDMs using Polynomial Regression. In *British Machine Vision Conference (BMVC'94)*, pages 397–405, 1994.

[104] P. Sozou, T. Cootes, C. Taylor, and E. Di-Mauro. Non-linear Point Distribution Modelling using a Multi-layer Perception. In *British Machine Vision Conference (BMVC'95)*, pages 107–115, Birmingham, UK, Sept. 1995.

[105] L. Staib and J. Duncan. Deformable Fourier models for surface finding in 3D images. In *Visualization in Biomedical Computing (VBC'92)*, pages 90–104, 1992.

[106] J. Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In M. Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 395–404. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.

[107] G. Storvik. A bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):976–986, 1994.

[108] G. Subsol, J.-P. Thirion, and N. Ayache. A scheme for automatically building three-dimensional morphometric anatomical atlases: application to skull atlas. *Medical Image Analysis*, 2(1):37–60, 1998.

[109] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5:271–301, 1990.

[110] R. Szeliski. Fast Surface Interpolation Using Hierarchical Basis Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513–528, June 1990.

[111] R. Szeliski and D. Tonnesen. Surface Modeling with Oriented Particle Systems. *Computer Graphics*, 26(2):185–194, July 1992.

[112] G. Székely, A. Kelemen, C. Brechbüler, and G. Gerig. Segmentation of 2D and 3D objects from MRI volume data using constrained elastic deformations of flexible Fourier surface models. In N. Ayache, editor, *International Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, volume 905 of *Lecture Notes in Computer Science*, pages 495–505, Nice (France), Apr. 1995. Springer Verlag.

[113] G. Székely, A. Kelemen, C. Brechbüler, and G. Gerig. Segmentation of 2D and 3D objects from MRI volume data using constrained elastic deformations of flexible Fourier surface models. *Medical Image Analysis*, 1(1):19–34, July 1996.

[114] G. Taubin. Distance approximation for Rasterizing implicit Curves. *ACM Transactions on Graphics*, 13(1):3–42, 1994.

[115] G. Taubin, C. Fernando, S. Sullivan, J. Ponce, and D. Kriegman. Parameterized Families of Polynomials for bounded Algebraic Curve and Surface Fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):287–303, 1994.

[116] D. Terzopoulos and K. Fleisher. Deformable models. *Visual Computer*, 4(6):306–331, 1988.

[117] D. Terzopoulos and D. Metaxas. Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.

[118] D. Terzopoulos, J. Platt, B. A., and K. Fleischer. Elastically deformable models. *ACM Computer Graphics (SIGGRAPH'87)*, 21(4):205–214, 1987.

[119] D. Terzopoulos and R. Szelisky. Tracking with Kalman snakes. In M. Press, editor, *Workshop on Active Vision*, pages 3–20, 1991.

[120] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[121] J.-P. Thirion. Image matching as a diffusion process: an analogy with Maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[122] M. Vasilescu and D. Terzopoulos. Adaptative meshes and shells. In *International Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pages 829–832, Champaign, Illinois, June 1992.

[123] B. Vemuri and Y. Guo. Snake Pedals : Geometric models with physic-based control. In *International Conference on Computer Vision (ICCV'98)*, pages 427–432, Bombay, India, Jan. 1998.

[124] B. Vemuri, Y. Guo, C. Leonard, and S.-H. Lay. Fast numerical algorithms for fitting multiresolution hybrid shape models to brain MRI. *Medical Image Analysis*, 1(4):343–362, 1997.

[125] B. Vemuri and A. Radisavljevic. From Global to Local, a Continuum of Shape Models with Fractal. In *International Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 307–313, New York, USA, 1993.

[126] B. Vemuri and A. Radisavljevic. Multiresolution Stochastic Hybrid Shape Models with Fractal Priors. *ACM Transactions on Graphics*, 13(2):177–207, 1994.

[127] F. Wehrmann, E. Bengtsson, and L. Thurfjell. A 3D Deformable Surface Model to Segment the Brain in MR Images. In *Scandinavian Conference on Image Analysis (SCIA'99)*, pages 289–294, Kangerlussuaq, Greenland, June 1999.

[128] D. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 55(1):14–26, Jan. 1992.

[129] A. Yuille and P. Hallinan. Deformable templates. In M. Press, editor, *Workshop on Active Vision*, pages 21–38, 1991.

[130] X. Zeng, L. Staib, R. Schlutz, and J. Duncan. Segmentation and Measurement of the Cortex from 3D MR Images. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'98)*, volume 1496 of *Lecture Notes in Computer Science*, pages 519–530, Cambridge, USA, Oct. 1998. Springer.

[131] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, Dec. 1994.

[132] D. Zorin, P. Schröder, and W. Sweldens. Interactive Multiresolution Mesh Editing. Technical Report CS-TR-97-06, Caltech, departement of computer science, Jan. 1997.

Contents

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399