A review of deformable surfaces: topology, geometry and deformation

J. Montagnat [1] , H. Delingette , and N. Ayache

INRIA, 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex, France
http://www-sop.inria.fr/epidaure/

Abstract

Deformable models have raised much interest and found various applications in the fields of computer vision and medical imaging. 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 [35,2], computer animation [100], geometric modelling [59], simulation [28], boundary tracking [11], image segmentation [69,67,91,5,45], etc. In this paper we propose a survey on deformable surfaces. Many surface representation have been proposed to meet different 3D reconstruction problem requirements. We classify the main representations proposed in the literature and we study the influence of the representation on the model evolution behavior, revealing some similarities between different approaches.

Key words: Deformable Surface, model representation, surface geometry, surface topology, 3D reconstruction

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 this research is to provide reliable reconstruction tools that are both robust and generic.

Deformable models cover a very wide range of applications and have been used in pattern recognition [35,2], computer animation [100], geometric modelling [59], surgery simulation [28], tracking [11], image segmentation [69,67,91,5,45],

---

[1] CREATIS, INSA 502, 20 bd A. Einstein, 69621 Villeurbanne Cedex, France
http://www.creatis.insa-lyon.fr/

etc. Extremely variable representations have been used to fulfill different 3D modelling needs, from deformable 3D lines [90,40] to deformable volumes [103,17]. 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 [46] and generalized to the 3D case by Terzopoulos *et al* [102]. Parametric representations such as superquadrics [99,7] and discrete representations [72,32] have also been proposed. Recently, implicit representations have been used with the ability to handle topology changes [58,112,53].

When reconstructing noisy 3D datasets [1], it is needed to constrain the model deformation and to introduce some prior information about the structure that is being recovered in the deformation process. For instance, statistical shape variations from a training set [26] may be used to constrain the deformation of a geometric model. In this paper, we propose a survey on existing deformable surfaces. We only focus on the geometric representation and evolution behavior of these surfaces without describing the numerous data forces that have been proposed since they are very dependent on the nature of the dataset (3D images, range images, unstructured point sets,...) and the target application.

Different approaches are classified according to their surface representations in section 2. The different schemes of deformation found in the literature are studied in section 3. Although there is a clear link between a surface representation and its evolution law, we exhibit some similarities between different approaches. Specific approaches for constraining the deformation of surface models have been proposed. These methods are described in section 4. The ability of various representations to handle topology changes is the topic of section 5.

The wide application field of deformable models has already led to the production of several surveys. McInerney and Terzopoulos [62] study deformable models for medical image analysis and classify the approach found in the literature based on their applications. Brown [14] focuses on registration methods as well as Audette *et al* [4] who thoroughly classify registration techniques from an algorithmic point of view. In this paper, we focus 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 can be characterized by its surface representation and its evolution law describing its ability to represent a different shape. We first focus on surface geometric representations.

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.* Instead of deforming directly a model shape, it is possible to deform its embedding space. For instance, a global transformation like an affine transformation may be applied to the model. The model number of degrees of freedom then depends on the class of allowed transformations. In the case of affine transformations, the deformation is only described by 12 parameters.

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 :

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

where (r, s) denotes a point of the surface parameter domain Ω.

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 :

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

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. 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 ($n_q \neq 0$).
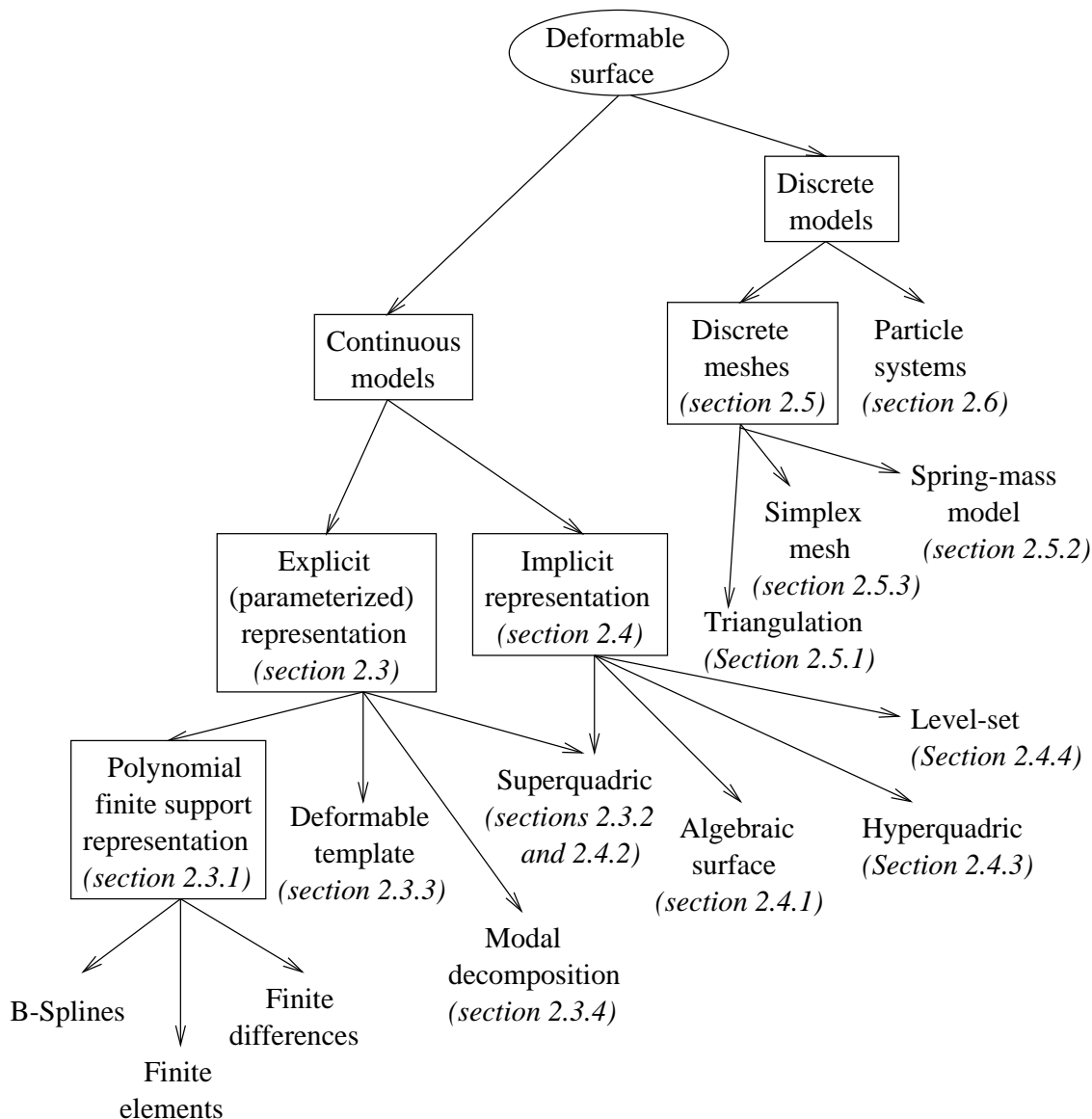
Fig. 1. Different geometric representations of deformable surfaces.

First, we make a distinction between deformable surfaces having continuous and discrete representations (see figure 1). 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 coordinate 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 of 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 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.

2.3.1   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 [64], 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 and contour tracking [40,10] in image sequences. Leitner *et al* [51,50] 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 [36]) 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.

2.3.2 Superquadrics

Superquadrics are a generalization of quadrics to surfaces. They certainly are the most widely used parameterized surfaces [99,7,80,106]. In particular, super-ellipsoids represent closed surfaces. A superellipsoid $\mathcal{Q}_{\mathbf{q}} : \Omega \to \mathbb{R}^3$ is defined by a parametric equation. The surface parameter vector $\mathbf{q}$ has only 6 components and $\Omega = [0, 2\pi[\times [-\frac{\pi}{2}, \frac{\pi}{2}]$. Superquadrics only allow the representation of shapes with a symmetry axis and are not suited to represent complex shapes.

Superquadrics can also be represented as implicit surfaces of a function $F_{\mathbf{q}} :$ $\mathbb{R}^3 \to \mathbb{R}$. The surface is then defined as $\mathcal{S}_{\mathbf{q}} = \left\{ \mathbf{p} \in \mathbb{R}^3 | F_{\mathbf{q}}(\mathbf{p}) = 1 \right\}$. This representation is suited to determine if a point $\mathbf{p}$ 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 also affects the evolution behavior of the model as we can see later in this paper.

2.3.3 Deformable templates

Yuille *et al* [111] 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.

6

2.3.4   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 [72,76].

Staib *et al* [87] investigate Fourier mode decomposition. A curve may be represented as a sum of sinusoidal terms. This representation extends to surfaces by using spherical harmonics, for instance, as in Székely *et al* [92]:

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

where Y_k^m is a spherical harmonic of degree k and order m, and $\mathbf{q} = (c_0^{-n}, \ldots, c_n^n)^T$ if only n modes are used. Other decomposition bases may be used. To simplify the surface parameterization, Staib and Duncan [87] 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 zero set of a function f valued in $\mathbb{R}$. That is:

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

2.4.1   Algebraic surfaces

Algebraic surfaces are surfaces satisfying equation 3 where f is a polynomial. They have been widely used in the field of graphics and modelling [96]. Taubin *et al* [97] 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.

2.4.2   Superquadrics

Superquadrics presented in section 2.3.2 may also be represented in an implicit form [7]. In particular, superellipsoids represent closed surfaces.

2.4.3   Hyperquadrics

Hyperquadrics [41,20] 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 remains homeomorphic to a sphere.

2.4.4   Level sets

Level set methods were developed by Osher and Sethian [74] and introduced in the medical vision community by Malladi *et al* [58]. Level set methods are thoroughly described in [83]. 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 $\Psi : \mathbb{R}^3 \to \mathbb{R}$: $\mathcal{S} = \{\mathbf{p} \in \mathbb{R}^3 | \Psi(\mathbf{p}) = 0\}$. Thus $\mathcal{S}$ is necessarily a closed surface or a set of closed surfaces.

Given an initial surface $\mathcal{S}_0$, the hypersurface 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 allow changes of surface topology implicitly. $\mathcal{S}$ 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* [57] propose several techniques to speed-up the numerical aspects including the use of narrow bands.

Siddiqi *et al* [84] 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 raises 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.

2.5.1 Triangulation

A discrete deformable contour $\mathcal{C}$ is just a set of vertices $\{\mathbf{p}_0, \ldots, \mathbf{p}_{d-1}\}$ connected to produce a closed or opened polygonal line. A surface is also represented by a set of vertices and a neighborhood relation that induces topological constraints. A common representation are triangulations (see left of figure 2) 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 neighboring triangle.

2.5.2 Spring-mass model

Vasilescu and Terzopoulos [104] use 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 [72] is shown in center of figure 2. 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 represent non-manifold surfaces with spring-mass models. Also, it may be necessary to add springs between vertices in order to improve the mesh stability and to provide a better volume behavior.

2.5.3 Simplex meshes

Simplex meshes were introduced by Delingette [32] and studied in [66,109,39,67]. 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 mesh is connected to three, and only three, neighbors (see right of figure 2). 2-simplex meshes are topologically dual to triangulations: 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 description allows the definition of shape memory constraints in the deformation process (see [32] for details).
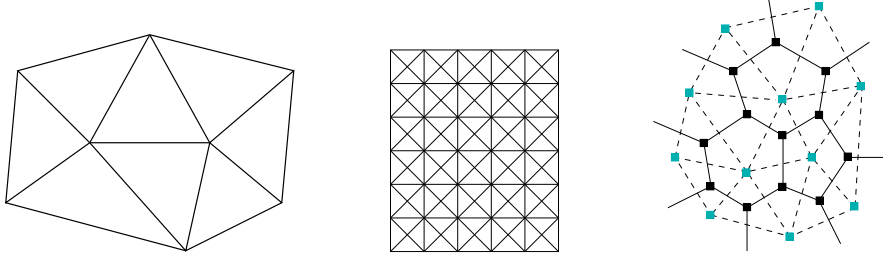

Fig. 2. Left: triangulation. Center: mass-spring model. Right: 2-simplex mesh and its dual triangulation.

2.5.4 Other discrete meshes

Vemuri and Guo [105] 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]. Subdivision surface techniques [44,88] 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 may lead 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. Particle systems are suited

to represent viscoelastic solids or fluids [34].

In [95], 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 [52] 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 [95] 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

For practical use, it is needed to deform the model to fit the data. The surface representation affects the possible evolution of a model. With few shape parameters, a surface model has restricted shape variations. Implicit representations allow topology changes but this property makes it also 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. External data is used to drive the model deformation but the presence of noise or the outliers requires to constrain the possible deformation space in order to stabilize the model evolution. Definition of the data term is very application dependent and falls out of the scope of this paper. The reader may refer to [23] for details.

Most deformable model evolution is driven the minimization of an energy functional E. This 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. The minimization of E often requires to differentiate and discretize the energy equation.

A different approach consists in avoiding the definition of a global energy functional and to directly propose a law of motion for the deformable model.

Indeed, the differential equation derived from the minimization of a global energy may lead to very complex complex expressions that are numerically difficult to implement. If we make an analogy with mechanical systems, the law of motion is then composed of a set of ad-hoc forces that do not derive from any energy. Powerful smoothing effect can be obtained by choosing appropriate internal forces also called *differential stabilizers* [31]. Figure 3 sums up the resolution methods described in this section.
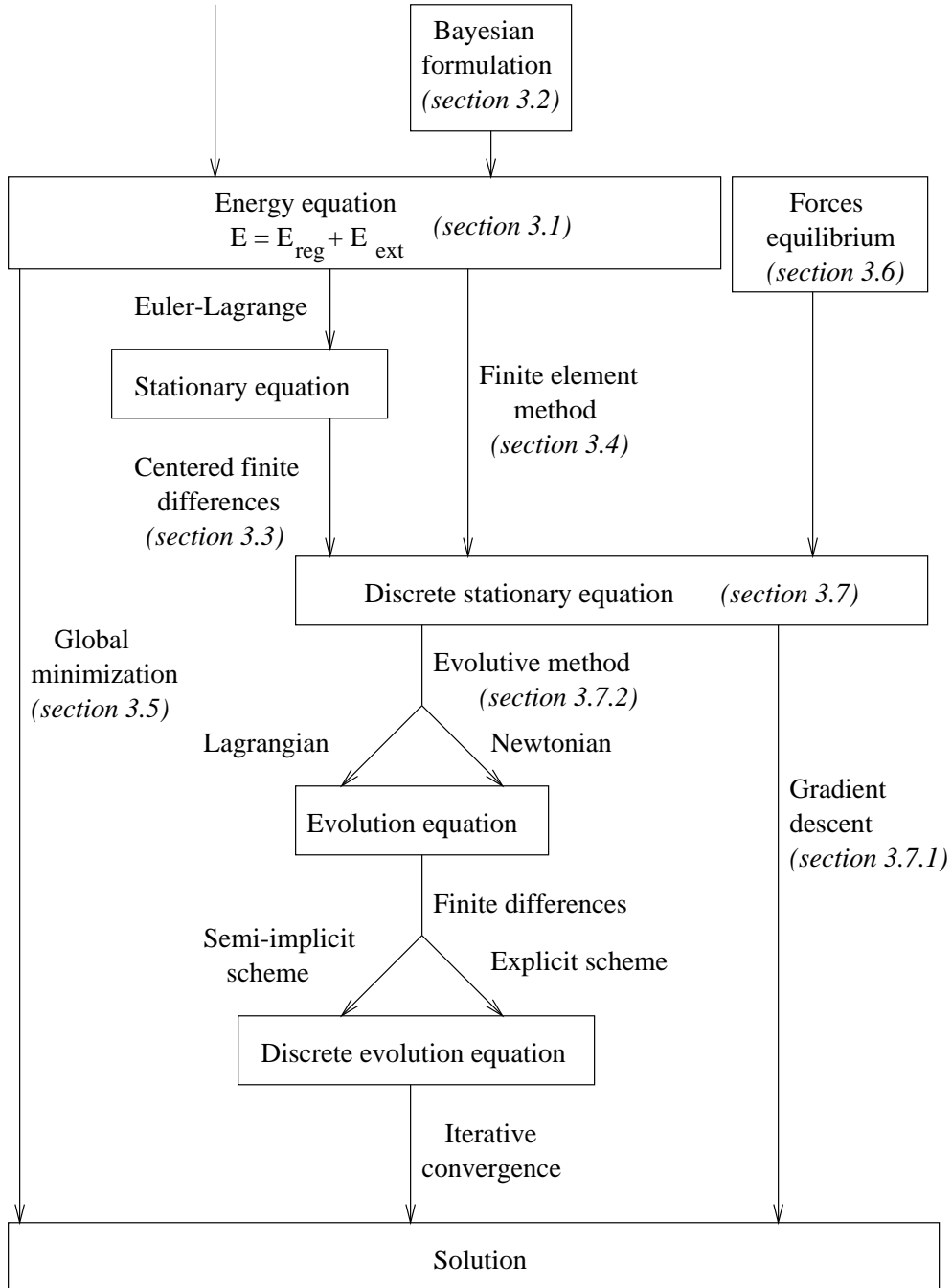
Bayesian formulation
(section 3.2)

Energy equation
$E = E_{reg} + E_{ext}$ *(section 3.1)*

Forces equilibrium
(section 3.6)

Euler-Lagrange

Stationary equation

Finite element method
(section 3.4)

Centered finite differences
(section 3.3)

Discrete stationary equation *(section 3.7)*

Global minimization
(section 3.5)

Evolutive method
(section 3.7.2)

Lagrangian   Newtonian

Evolution equation

Gradient descent
(section 3.7.1)

Finite differences

Semi-implicit scheme   Explicit scheme

Discrete evolution equation

Iterative convergence

Solution

Fig. 3. Solving deformable models' evolution equation.

12

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: $L^2(\Omega) = \{f \in (\Omega \to \mathbb{R}) \,|\, \int_\Omega f^2 < +\infty\}$. $L^2(\Omega)$ is characterized by the scalar product $(f, g) = \int_\Omega fg$ and the induced norm $\|f\|^2 = (f, f)$.

The energy of a parameterized surface $\mathcal{S} \in L^2(\Omega)^3$ is defined by:

$$E : L^2(\Omega)^3 \to \mathbb{R}$$

$$\mathcal{S} \mapsto E(\mathcal{S}) = E_{\text{reg}}(\mathcal{S}) + E_{\text{ext}}(\mathcal{S})$$

We admit that the energy functional is built such that its global minimum coincide with the expected solution $\mathcal{S}_0 = \min_{\mathcal{S}} E(\mathcal{S})$. As a regularizing term, Tikhonov stabilizers are often used to measure the surface smoothness. Thus, a surface internal energy is:

$$E_{\text{reg}}(\mathcal{S}) = \int_\Omega \sum_{1 \le i+j \le k} \frac{(i+j)!}{i!j!} w_{ij}(r, s) \left\| \frac{\partial^{i+j}\mathcal{S}}{\partial r^i \partial s^j} \right\|^2 dr\,ds \qquad (4)$$

where w_{ij} are positive functions. Second order stabilizers ($k = 2$) are mostly used. 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. A surface regularized by a second order Tikhonov stabilizer is identical to a thin plate 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): $E_{\text{ext}}(\mathcal{S}) = \int_\Omega P(\mathcal{S}(r, s)) dr\,ds$.

For given E_{reg} and E_{ext} terms, the minimization equation does not have, in general, any analytical solution. Thus, it is necessary 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 [110] or simulated annealing [77,89].

Another classical approach to optimize the shape of a surface $\mathcal{S}$ embedded in a dataset $\mathcal{D}$ is to use a statistical framework [111,93]. Probability $P(\mathcal{S}|\mathcal{D})$ denotes the *posterior statistical model*. It estimates how a surface $\mathcal{S}$ fits the dataset $\mathcal{D}$. The searched solution is the *posterior maximum* (MAP), that is the surface that maximizes the *posterior* model probability MAP $= \max_{\mathcal{S}} P(\mathcal{S}|\mathcal{D})$.

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

$$P(\mathcal{S}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{S})P(\mathcal{S})}{P(\mathcal{D})} \tag{5}$$

composed of the following terms:

- $P(\mathcal{D}) = \sum_{\mathcal{S}} P(\mathcal{D}|\mathcal{S})$ 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* $P(\mathcal{D}|\mathcal{S})$ measures how the given surface generates the dataset. This is the stochastic process description linking the unknown state $\mathcal{S}$ with the data $\mathcal{D}$.

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: $P(\mathcal{D}|\mathcal{S}) = \frac{1}{Z_{\text{ext}}}e^{-E\text{ext}(\mathcal{S})}$. Finally, the *posterior* model is: $P(\mathcal{S}|\mathcal{D}) = \frac{1}{Z}e^{-E(\mathcal{S})}$. Thus, the maximum of the conditional probability $P(\mathcal{S}|\mathcal{D})$ 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 [93].

Terzopoulos and Szeliski [101] 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 with several local minima. The goal of the minimization procedure is to push the model towards one of those

minima. The minimization result thus depends on the initial pose and shape of the model.

3.3.1 Stationary equation

Using Euler-Lagrange equation ($\nabla E(\mathcal{S}) = 0$), and considering a second order Tikhonov stabilizer, it can be proved [19] that an energy local minimum must satisfy:

$$-\frac{\partial}{\partial r} w_{10} \frac{\partial \mathcal{S}}{\partial r} - \frac{\partial}{\partial s} w_{01} \frac{\partial \mathcal{S}}{\partial s} + 2\frac{\partial^2}{\partial r \partial s} w_{11} \frac{\partial^2 \mathcal{S}}{\partial r \partial s} + \frac{\partial^2}{\partial r^2} w_{20} \frac{\partial^2 \mathcal{S}}{\partial r^2} + \frac{\partial^2}{\partial s^2} w_{02} \frac{\partial^2 \mathcal{S}}{\partial s^2} = -\nabla P(\mathcal{S}) \quad (6)$$

plus boundary conditions. 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 $H^4(\Omega)^3$ space. A solution to equation 6 can be computed by discretizing this equation using centered finite differences.

3.3.2 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 such as:

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

After discretizing equation 4, 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: $\mathbf{K}(\mathcal{S})\mathcal{S} = \mathbf{f}(\mathcal{S})$. When using Tikhonov stabilizers, if w_{ij} functions are constants, the system simplifies in a linear form :

$$\mathbf{K}\mathcal{S} = \mathbf{f}(\mathcal{S}). \tag{7}$$

3.4 Discretization using finite elements

It can be proved [19] that solving equation 6 is equivalent to finding a function $v : t \in [0, T] \mapsto v(t) = (\mathbf{u} \in \Omega \mapsto \mathcal{S}(\mathbf{u}, t)) \in H_0^2(\Omega)$, where $H_0^2(\Omega)$ is the

Sobolev space, such as $\forall u \in H_0^2(\Omega), a(v(t), u) = L_v(u)$, where a is a bilinear form defined as:

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

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 $\left\{p(r, s) = \sum_{0 \le k, l \le 3}\gamma_{k,l}s^k r^l\right\}$. $V_h = \left\{v \in \mathcal{C}^1(\Omega)|v_{|K_{i,j}} \in Q_3(K_{i,j})\right\}$ 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 $\mathbb{R}$. 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 7.

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 [23].

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 [89] derives from the gradient descent method. Starting from an initial surface $\mathcal{S}_0$, a gradient descent iteratively evolves according to the equation $\mathcal{S}_{k+1} = \mathcal{S}_k - \Delta t \nabla \mathbf{E}(\mathcal{S}_k)$ until the model converges. The simulated annealing process 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 $\Delta \mathbf{E_q} = \mathbf{E}_{\{q_0,\dots,q_i+\xi,\dots,q_n\}} - \mathbf{E_q}$ 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 $e^{\frac{\Delta \mathbf{E_q}}{\tau}}$ where τ 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 $\mathcal{C} = \{\mathbf{p}_0, \dots, \mathbf{p}_{d-1}\}$. This energy can be written:

$$E(\mathcal{C}) = E_0(\mathbf{p}_0, \mathbf{p}_1) + E_1(\mathbf{p}_1, \mathbf{p}_2) + \dots + E_{d-1}(\mathbf{p}_{d-1}, \mathbf{p}_{d-2}). \tag{8}$$

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* [38] propose an optimization of 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 pair of vertices. The algorithm might then fail if no solution satisfies the given hard constraints. However, this algorithm extension to the surface case is made difficult by the complex decomposition of the energy functional as a sum equivalent to equation 8.

3.6 Forces equilibrium

Each term of the stationary equation 6 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 [33]. The optimal model then verifies the equilibrium equation $\sum f_{\text{reg}}(\mathcal{S}) = -\sum f_{\text{ext}}(\mathcal{S})$ which can be solved in a similar way to the energy stationary equation 6.

Since these forces do not derive from any 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 regularizing constraint associated with the internal forces can be identified by looking at the model shapes where these forces vanish. The external forces must guide the model towards the data. To avoid any 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. Another approach is to introduce an evolution 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 [78,56] 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 or finite element scheme as detailed in section 3.3 for instance. The simple gradient descent 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:

$$\mathcal{S}_{k+1} = \mathcal{S}_k - \Delta t \nabla \mathbf{E}(\mathcal{S}_k) \tag{9}$$

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 large number of iterations. Therefore, faster convergence numerical schemes such as conjugate gradient, Newton method or Levenberg-Marquardt method have been proposed [78].

In [106], Vemuri *et al* introduce new numerical algorithms and compare their efficiency.

3.7.2 Evolutive approach

The discrete stationary equation 7 is not linear. Its resolution requires an evolutive framework that iterates towards the solution. Many methods for solving iterative systems are proposed in [78]. The evolutive approach consider the deformable model as a dynamic object evolving according to a parameter t [98]:

$$\mathcal{S} : \Omega \times [0, +\infty] \rightarrow \mathbb{R}^n$$

$$(\mathbf{u}, t) \mapsto \mathcal{S}(\mathbf{u}, t)$$

and the corresponding energy term $\mathbf{E}(\mathcal{S}, t) = \mathbf{E}(\mathcal{S}(., t))$.

Inertial terms are introduced in equation 7, of the first order (Lagrangian evolution) $\frac{\partial \mathcal{S}}{\partial t} + \mathbf{K}\mathcal{S} = \mathbf{f}(\mathcal{S})$ or of the second order (Newtonian evolution) :

$$\mathbf{M}\frac{\partial^2 \mathcal{S}}{\partial t^2} + \mathbf{C}\frac{\partial \mathcal{S}}{\partial t} + \mathbf{K}\mathcal{S} = \mathbf{f}(\mathcal{S}), \tag{10}$$

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 evolution 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. [99,22].

The dynamic system 10 must be discretized in time by the finite difference scheme. $\mathbf{M}$ 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:

$$\mathbf{M}\frac{\mathcal{S}_{t+\Delta t} - 2\mathcal{S}_t + \mathcal{S}_{t-\Delta t}}{\Delta t^2} + \mathbf{C}\frac{\mathcal{S}_{t+\Delta t} - \mathcal{S}_t}{\Delta t} + \mathbf{K}\mathcal{S}_t = \mathbf{f}(\mathcal{S}_t), \tag{11}$$

converges only if the selected time step is small enough [78]. However, equation 11 give an explicit way of computing $\mathcal{S}_{t+\Delta t}$ from $\mathcal{S}_t$ and $\mathcal{S}_{t-\Delta t}$. This equation is simple and can be solved at low cost. If $\mathbf{M} = 0$ and $\mathbf{C} = \mathbf{I}$, it simplifies to $\mathcal{S}_{t+\Delta t} = \mathcal{S}_t - \Delta t(\mathbf{K}\mathcal{S}_t - \mathbf{f}(\mathcal{S}_t)) = \mathcal{S}_t - \Delta t \nabla \mathbf{E}$. It can be seen that the Lagrangian explicit numerical scheme leads to an equation identical to a gradient descent 9.

Alternatively, a semi-implicit scheme can be used:

$$\mathbf{M}\frac{\mathcal{S}_{t+\Delta t} - 2\mathcal{S}_t + \mathcal{S}_{t-\Delta t}}{\Delta t^2} + \mathbf{C}\frac{\mathcal{S}_{t+\Delta t} - \mathcal{S}_{t-\Delta t}}{2\Delta t} + \mathbf{K}\mathcal{S}_{t+\Delta t} = \mathbf{f}(\mathcal{S}_t). \tag{12}$$

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 12 requires the resolution of a dependent equation set at each iteration. 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.

3.8   Discrete model evolution

A discrete model geometry is defined only at its vertices. A vertex i might be considered as a point mass governed by a Newtonian law of motion:

$$m(i)\frac{d^2\mathbf{p}_i}{dt^2} = -\gamma(i)\frac{d\mathbf{p}_i}{dt} + \alpha(i)f_{\text{reg}}(\mathbf{p}_i) + \beta(i)f_{\text{ext}}(\mathbf{p}_i), \qquad (13)$$

where $m(i)$ is the vertex mass, $\gamma(i)$ is the background damping, f_{reg} 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 10. 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.

3.9   Implicit models evolution

3.9.1   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 [80,96].

3.9.2   Level-set methods

Malladi *et al* [58] qualify as *Eulerian dynamic* the level-set evolution equations by opposition to the *Lagrangian dynamic* of "classical" deformable models.

21

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:

$$\frac{\partial \Psi}{\partial t} = \bar{\nu}(\mathbf{u}, t)|\nabla \Psi| \qquad (14)$$

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 14 can be solved by using finite differences (in space and time):

$$\Psi_{ijk}^{t+\Delta t} = \Psi_{ijk}^t + \Delta t \bar{\nu}_{ijk} |\nabla_{ijk} \Psi_{ijk}^t| \qquad (15)$$

Moreover, the surface geometrical attributes are determined by the analytical expressions $\mathbf{n} = \frac{\nabla \Psi}{|\nabla \Psi|}$ and $H = \nabla \cdot \frac{\nabla \Psi}{|\nabla \Psi|}$.

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 15 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 10 or 13 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 techniques only lead to suboptimal 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.

4.1.1 Iterative Closest Point algorithm

The registration process described by Besl and McKay [9] and Zhang [113] is very close to the iterative resolution of the deformable models dynamic equation. A model is iteratively deformed by a global transformation T : $\mathbb{R}^3 \to \mathbb{R}^3$. The transformation T is estimated by minimizing the least square criterion:

$$T = \arg \min_{T \in \mathrm{Treg}} \sum_{i=0}^{d-1} \|T(\mathbf{p}_i) - (\mathrm{Closest}(\mathbf{p}_i))\|^2 \tag{16}$$

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).

4.1.2 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 16 [75]. Rigid registration is useful for aligning data originating 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.

4.1.3 Free-form deformations and B-spline transformations

Sederberg and Parry [82] and Coquillart [27] 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.

Declerck *et al* [30] 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* [55,54] and Bardinet *et al* [7] similarly need to add a regularizing term to FFD.

4.1.4 Superquadric extensions

Terzopoulos *et al* [99] 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* [65] 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 $\mathbf{q} = (\mathbf{q}_c, \mathbf{q}_\theta, \mathbf{q}_l, \mathbf{q}_d, \mathbf{q}_T)$.

4.1.5 Other globally constrained deformations

Feldmar and Ayache [37] define locally affine transformations to increase the surface 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 trans-

formation and demonstrate some continuity results on the surface patches.

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 [12] 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 [29] similarly build a functional family varying with a scalar parameter to drive a thick line detector model. Cohen and Gorre [24] 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 data point.

Montagnat and Delingette [68] regularize deformations through a modified deformation scheme that combines local and global forces resulting from a global transformation estimation. By weighting the global and the local forces, it is possible to continuously evolve from a global registration process to a local deformation one.

4.2.2 Pyramidal approach

Reissman et al [79] 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 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 [55] use 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 [94] 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 [37] 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 [67] 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 [87]), give an incrementally detailed representation of surfaces.

Pentland *et al* [76] similarly optimize the resolution of the model evolution 10 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 10 rewrites:

$$\tilde{\mathbf{M}}\frac{d^2\tilde{\mathcal{S}}}{dt^2} + \tilde{\mathbf{C}}\frac{d\tilde{\mathcal{S}}}{dt} + \tilde{\mathbf{K}}\tilde{\mathcal{S}} = \tilde{\mathbf{f}}, \tag{17}$$

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 $\mathbf{\Phi}$ is the common eigenmodes matrix of $\mathbf{M}$, $\mathbf{K}$ and $\mathbf{C}$ such that components of equation of 17 are independent. $\mathbf{\Phi}$ is the solution to the double eigenproblem $\mathbf{K}\mathbf{\Phi} = \mathbf{\Omega}^2\mathbf{\Phi}\mathbf{M}$, 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 $\mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi} = \mathbf{\Omega}^2$ and $\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi} = \mathbf{I}$. The eigenvector ϕ_i is the ith 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 series [8]: $\mathbf{C} = \mathbf{M}\sum_{k=0}^{p-1} a_k[\mathbf{M}^{-1}\mathbf{K}]^k$. If $p = 2$ then $\mathbf{C} = a_0\mathbf{M} + a_1\mathbf{K}$ and equation 17 simplifies to $\frac{d^2\tilde{\mathcal{S}}}{dt^2} + \tilde{\mathbf{C}}\frac{d\tilde{\mathcal{S}}}{dt} + \mathbf{\Omega}^2\tilde{\mathcal{S}} = \mathbf{\Phi}^T\tilde{\mathbf{f}}$, 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 [72] use Pentland and Sclaroff's modal analysis to reduce the model deformation space. Bookstein [13] similarly decomposes a thin-plate spline deformations over the eigenmodes of the bending energy matrix. Cohen and Cohen [21] 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 [99], Vemuri and Radis-avljevic [107] propose a continuous transition between global deformations (only involving variations of the superquadric parameters) and local deformations (involving the **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* [25] 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: $\mathbf{x}_i = (x_1, \ldots, x_d)^T$ [6,26]. In that case, T is a similarity transformation. Vector $\mathbf{x}_i$ may also be a parameter vector: $\mathbf{x}_i = \mathbf{q} = (q_1, \ldots, q_d)^T$ [91]. 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 $\mathbf{V} = \mathrm{diag}(v_1, \ldots, v_n)$ 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 for a human operator except for a few feature points such as curvature extrema [18]. Anyway, several methods to automatically match different shapes are proposed by Caunce and Taylor [16].

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

Cootes *et al* [26] 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 $\mathbf{dx} = \mathbf{E}_{t+\Delta t}^{-1}\mathbf{R}_{t+\Delta t}^{-1}(\mathcal{S}_t + d\mathcal{S} - \mathbf{t}_{t+\Delta t}) - \mathbf{x}_t$ with $\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \mathbf{dx}$.

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 [26], in particular segmentation of anatomical structures [81,43], contour tracking [42,47], and shape classification [73]. Some non linear extensions have been proposed in [86,85].

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 at left of figure 4) or change of the surface genus (right of figure 4) 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.
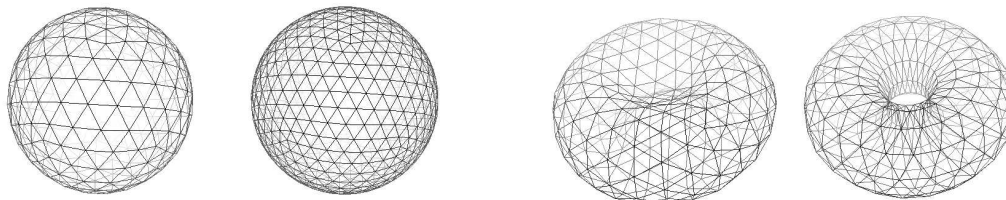

Fig. 4. Left: adaptive topology. Right: topology change.

5.1 *Adaptive models*

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

5.1.1 *Parameterized models adaptation*

Parameterized models require a data adaptation if the parameter space discretization is rough compared to the data detail level. McInerney and Terzopoulos [60] 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 parameter space does not necessarily lead to a regular sampling of the surface. Vemuri and Radisavl-jevic [108] propose a transformation of parameter space solving the problem in their particular case. Székely *et al* [91] try 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 numerical scheme, these matrices have to be inverted. An adaptive scheme with too high a frequency might thus become computationally expensive.

5.1.2   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 [15] propose a refinement algorithm for regular triangulated surfaces (each vertex has six neighbors). Delingette [32] 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 $\{\mathbf{p}_i^0\}_i$ 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 [114]. A second approach is to consider the points $\{\mathbf{p}_i^0\}_i$ as a control node set indirectly generating a surface. The surface is refined by adding control points.

5.1.3   Adapting the number of modes

Decomposing a model on a vibration mode basis as described in [70,71,108,76] 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 [48,63,49]. 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, it is difficult to apply when considering noisy datasets. The deformable models convergence is very dependent on its prior shape and location. To automatically determine topology changes in a noisy dataset without prior shape information may be unstable.

5.2.1   Parameterized representations

When considering parameterized models, different topologies lead to different parameter boundary conditions. Leitner and Cinquin [50] 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.

5.2.2   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.

5.2.3   Discrete representation

McInerney and Terzopoulos [61] 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. The algorithm alternatively performs deformations and resampling. 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 [48] define topology adaptive triangulations based on the δ-snakes concept. A length parameter ($\delta > 0$) is used to control the triangulation sampling and to detect model self-intersections from two elementary rules controlling the distance between neighbor vertices and non neighbor vertices. 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.

Delingette and Montagnat [33] also propose an algorithm for handling topology changes of discrete contours. A contour subsampling over a grid is used to speed-up intersection detections.

6   Conclusion

This survey focuses on the topological, geometric and evolution aspects of deformable models because they are common to all deformable models independently of their applications. However, it is clear that we have not covered other aspects of deformable models that are typically application-dependent. For instance, when considering deformable models in the context of image segmentation, the choice related to external energy guiding the models towards an object boundary is also of primary importance. In computer graphics, the issues of collision detection and realistic behavior are more relevant. These specific topics have been partially covered by previous surveys [62].

In section 2 we have classified various surfaces by their geometric representation. The choice of a surface representation has an effect on the set of possible shapes. For instance, surfaces defined as sums of spherical harmonics are restricted to represent shapes of genus 0. Superquadrics can only represent convex objects. Level-sets cannot represent contours or surfaces with borders. Conversely, discrete meshes can represent surfaces with any topology. Some representations are continuous and allow to define geometric quantities such as normals or curvatures all over the surface. Conversely, discrete representations only define the surface geometric attributes at a finite set of points. The representation is thus very context-dependent, depending on the desired shape variability and geometric properties.

Deformation schemes are the topic of section 3. The problem of deforming a surface towards a dataset is often stated as an optimization problem that

involves the minimization of a model energy functional. The energy functional both depends on the surface regularity and the distance between the surface and the data. In the ideal case, the optimal surface is a global minimum of the energy functional that corresponds to an equilibrium between regularity and distance to data. Few techniques allow the search of a global minimum for the very irregular and non-convex energy functional because of their computational complexity. Instead, the energy equation is often derived in a stationary equation corresponding to an equilibrium of forces applied onto the surface. Local minimization of the derived equation is achieved either by a gradient descent or an evolutive method. The surface is thus likely to converge towards a local minima of its energy.

Section 4 presents several techniques to constrain the deformation space. Constrained deformations lead to more convex energy functional that are easier to minimize. They are often used to reduce the surface degrees of freedom in the initialization stage. However, constraining deformations too much may reduce the surface shape variability and make it impossible to reconstruct complex 3D structures. A trade-off between deformation behavior and shape variability must be found. Iterative or multi-scale algorithms have been proposed to apply global, highly constrained, transformations and evolve towards local deformation as the algorithm converges. Conversely, some surface representations have very few degrees of freedom. Superquadrics, for instance, are only defined by a set of six parameters. Additional local deformations may be used to improve the surface space of deformations.

Finally, we have focused on model topology changes in section 5. Topology changes include both surface adaptation and changes in the surface genus. Implicit representations such as the level-set algorithm naturally allow topological adaptation. For explicit representation, few sophisticated algorithms have been proposed to allow such adaptivity. These algorithms are often based on a smart sampling of the Euclidean space.

Acknowledgments

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.

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

[3]   A.A. Amini, T.E. Weymouth, and R.C. 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.A. Audette, F.P. Ferrie, and T.M. Peters. An Algorithmic Overview of Surface Registration Techniques for Medical Imaging. *Medical Image Analysis*, 4(3):201–217, September 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 Taylor, Lavallée, Burdea and Mösges, editor, *Computer Integrated Surgery, Technology and Clinical Applications*, pages 59–74. MIT press, 1996.

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

[7]   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.

[8]   K. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall, 1982.

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

[10]   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.

[11]   A. Blake and M. Isard. 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.

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

[13]   F.L. 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.

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

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

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

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

[18] 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, August 1997.

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

[20] 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.

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

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

[23] L.D. 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, November 1993.

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

[25] T.F. Cootes, A. Hill, C.J. 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 *LNCS*, pages 33–47, Flagstaff, USA, June 1993. Springer-Verlag.

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

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

[28] S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *Visual Computer*, 16(8):437–452, 2000.

35

[29] 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.

[30] J. Declerck, J. Feldmar, M.L. 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, December 1997.

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

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

[33] H. Delingette and J. Montagnat. New algorithms for controlling active contours shape and topology. In *European Conference on Computer Vision (ECCV'00)*, pages 381–395, Dublin, Ireland, June 2000.

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

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

[36] M. Eck and H. Hoppe. Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type. In *Computer Graphics (SIGGRAPH'96)*, 1996.

[37] 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.

[38] D. Geiger, A. Gupta, A.C. 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.

[39] M.A. 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 *LNCS*, pages 499–508, Cambridge, USA, October 1998. Springer.

[40] 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.

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

[42] 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, January 1998.

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

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

[45] 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, September 1995.

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

[47] 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.

[48] 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.

[49] F. Leitner and P. Cinquin. Complex Topology 3D Objects Segmentation. In *SPIE Conf. on Advances in Intelligent Robotics Systems*, volume 1609, Boston, November 1991.

[50] 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 *LNCS*, pages 264–281, Grenoble, France, 1993.

[51] F. Leitner, I. Marque, 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.

[52] Jean-Christophe Lombardo. *Modélisation d'objets déformables avec un système de particules orientées*. PhD thesis, Université Joseph-Fourier, iMAGIS/IMAG, Grenoble, France, January 1996.

[53] L. Lorigo, O. Faugeras, W.E.L. 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 *LNCS*, pages 126–139, Visegrád, Hungary, June 1999. Springer.

[54] 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 *LNCS*, pages 192–201, Cambridge, UK, September 1999. Springer.

[55] 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.

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

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

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

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

[60] 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 *LNCS*, pages 518–523. Springer-Verlag, April 1995.

[61] 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 *LNCS*, pages 92–100. Springer-Verlag, April 1995.

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

[63] 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 *LNCS*, pages 92–100. Springer-Verlag, March 1997.

[64] 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.

[65] 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.

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

[67] 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 *LNCS*, pages 13–22. Springer-Verlag, March 1997.

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

[69] 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 *LNCS*, pages 176–183, Cambridge, UK, September 1999. Springer.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

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

[75] 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.

[76] 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.

[77] 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, October 1994. SPIE.

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

[79] 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.

[80] 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.

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

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

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

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

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

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

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

[88] J. Stam. Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values. In *ACM Computer Graphics (SIGGRAPH'98)*, pages 395–404, July 1998.

[89] 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.

[90] 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.

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

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

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

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

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

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

[97] G. Taubin, C. Fernando, S. Sullivan, J. Ponce, and D.J. 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.

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

[99] 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.

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

[101] D. Terzopoulos and R. Szeliski. *Active Vision, chapter Tracking with Kalman Snakes*. MIT Press, 1992.

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

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

[104] 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.

[105] B.C. 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, January 1998.

[106] B.C. Vemuri, Y. Guo, C.M. 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.

[107] B.C. 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.

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

[109] 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.

[110] D.J 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, January 1992.

[111] A. Yuille and P. Hallinan. *Active Vision, chapter Deformable templates*. MIT Press, 1992.

[112] 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 *LNCS*, pages 519–530, Cambridge, USA, October 1998. Springer.

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

[114] D. Zorin, P. Schröder, and W. Sweldens. Interactive Multiresolution Mesh Editing. In *ACM Computer Graphics (SIGGRAPH'97)*, pages 259–268, 1997.