Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images

Laurent D. Cohen and Isaac Cohen

***Abstract*— The use of energy-minimizing curves, known as "snakes" to extract features of interest in images has been introduced by Kass, Witkin and Terzopoulos [23]. A balloon model was introduced in [12] as a way to generalize and solve some of the problems encountered with the original method. A 3-D generalization of the balloon model as a 3-D deformable surface, which evolves in 3-D images, is presented. It is deformed under the action of internal and external forces attracting the surface toward detected edgels by means of an attraction potential. We also show properties of energy-minimizing surfaces concerning their relationship with 3-D edge points. To solve the minimization problem for a surface, two simplified approaches are shown first, defining a 3-D surface as a series of 2-D planar curves. Then, after comparing finite-element method and finite-difference method in the 2-D problem, we solve the 3-D model using the finite-element method yielding greater stability and faster convergence. This model is applied for segmenting magnetic resonance images.**

***Index Terms*—Active contour models, attraction potential, deformable models, feature extraction, finite difference method, finite element method, regularization, segmentation, surface reconstruction**

I. Introduction

We study segmentation of medical 2-D and 3-D images by making use of "deformable models" [29], [32] In order to achieve robust segmentation, we introduce a number of enhancements and modifications to the formulation of deformable models. In particular, we define new forces to control the evolution of the deformable model, we formulate the models for true 3-D data, and we develop a finite-element implementation.

The class of "deformable models" originates with the method of "snakes" introduced by Kass *et al.* [23], which are used to locate smooth curves in 2-D imagery. Since then, deformable models have been used for many applications in $2\frac{1}{2}$-D and 3-D by Terzopoulos, Witkin and Kass [31], [32] where the deformable surface is constrained to encourage axial symmetry and is evolving under the forces determined from a 2-D image or a pair of 2-D images. We also make use of deformable surfaces, but the data providing information about the force comes from *true* 3-D *data* sets. We further extend enhancements of the model introduced in [12] for curves to the surface model applications given here.

In [12], we introduced a modification, using "balloons," in order to apply the method of deformable models to stacks of images comprising a 3-D data set for an application in segmentation. Our use of deformable models in [12] was limited to the extraction of 2-D curves, which were then used to build up a 3-D structure. In this paper, we further refine and present the "balloon model," formulating and applying it to true 3-D data. For this purpose, we study the use of finite-element methods for implementing the solution of the partial differential equations satisfied by the deformable surface. Our application is for the segmentation of 3-D magnetic resonance images of crania and heart regions.

We compare different schemes using finite-difference and finite-element methods to generalize the balloon model introduced in [12] to a 3-D cylindrical surface or rectangular patch. In general, these methods are used to reliably extract surfaces in 3-D images.

Three-dimensional imagery is often represented as a set of intensity voxels (volume elements). A 3-D edge detector, after a *local* image analysis [36], [24], provides a set of 3-D edgels (edge elements). However, the edgels do not constitute a segmentation. One approach to 3-D segmentation involves the integration of 2-D segmentation results along slices of the 3-D imagery. In this paper, however, we wish to combine information from a 3-D edge detector with the method of deformable models applied directly to the solid data.

We are confronted simultaneously with a segmentation problem and a surface reconstruction problem:

1) we wish to locate edgels belonging to the surface of a single object; this is the segmentation problem; and
2) we must represent the surface, together with its differential structure, for subsequent interpretation [1].

Deformable models offer a reasonable approach to solving these problems, due to their stability, controllability, and their property of regularizing data gathered over regions of the image. Regularization techniques, or penalized optimization, are used for many applications in vision (see for example [21], [28], [27], [30] and references there).

In our application, we recover surfaces in 3-D medical data, locating surface boundaries of organs and structures, and providing an approximating differentiable description (see Section VI). The differential description may be used for measurements, recognition, visualization, and other purposes [2], [10], [13].

There are two basic approaches to segmentation and image

Manuscript received November 14, 1991; revised December 10, 1992. This work was supported in part by the Digital Equipment Corporation. Recommended for acceptance by Associate Editor T. Henderson.

The authors are with CEREMADE, U.R.A. CNRS 749, Université Paris IX-Dauphine, Place du Marechal de Lattre de Tassigny, 75775 Paris CEDEX 16, France and INRIA, Domaine de Voluccau, Rocquencourt B.P. 105, 78153 Le Chesnay CEDEX, France.

IEEE Log Number 9212248.

Classic: $\rightarrow$ $\boxed{\text{sparse feature extraction}}$ $\rightarrow$ $\boxed{\text{interpolative reconstruction}}$ $\rightarrow$

Snakes: $\rightarrow$ $\boxed{\text{optimization in image domain}}$ $\rightarrow$

Our model: $\rightarrow$ $\boxed{\text{edge extraction}}$ $\rightarrow$ $\boxed{\text{optimization}}$ $\rightarrow$

Fig. 1. Comparison of reconstruction approaches. Constraints are *explicit* in the first model, *implicit* in the others.

labeling (see Fig. 1). In the classical approach, features are extracted from the image, and a sparse collection of locations and data are obtained; then reconstruction methods are used to interpolate the sparse data to form a representation (and possible segmentation) of the original data. In more recent approaches, such as the method of snakes [23], an initial estimate (such as a curve or surface) is provided, and optimization methods are used to refine the initial estimate based on image data restricted to the region of the evolving estimate. The second approach has the advantage that the feature extraction and representation phases are integrated into a single process, whereas the first approach may make use of prior finely-tuned feature extraction procedures.

In our work, we modify the second approach by incorporating aspects of the first, namely, the evolution of the initial estimate depends not only on local data, but also potentially on the data provided by a distributed sparse collection of feature points such as edgels from a surface edge extractor. Our method, which can make use of a "inflation" or "weight" force, is particularly well suited to noisy data with missing parts such as magnetic resonance images in both two and three dimensions.

Our method is derived from the original formulation of deformable models [23, 29], but incorporates a number of significant modifications and new features.

In particular, the contributions of our work are the following:

1) We incorporate the use of *edge points* extracted by a local edge detector. This allows us to combine the qualities of a good *local* edge detector, e.g., a Canny–Deriche edge extractor [7], [17], [24], with a *global* active model. This is accomplished by means of an attraction potential generated by convolving a binary edge image with a Gaussian impulse response. The attraction potential can also be defined through the use of a Chamfer distance to edge points.
2) We introduce an internal pressure force by regarding our curve or surface as a *balloon* which is inflated. We add to the previous internal and external forces a *pressure* force pushing out the boundary as if we were introducing air inside. Separately, we make use of a "*weight*" force which simulates gravity. This allows us to be less demanding of the initialization and to give a simpler initial curve or surface.
3) We replace the finite-difference method of [12], [23], [30] by a *finite-element method* (FEM). With finite differences, we only have knowledge of the functions at discrete points of a subdivision, and have no information between these points. Therefore, the distance between successive points must be made very small to achieve sufficient precision so as not to miss too much information, since the external forces are applied at the grid of points. This typically yields large systems of linear equations. Conversely, with the FEM, we work with continuous functions whatever the size of the grid. Therefore, the function under consideration is known everywhere in the image, independently of the chosen discretization. This yields a lower algorithmic complexity and better numerical stability, in our application.
4) We deal with true 3-D medical data and use deformable surfaces to extract the surface boundary of organs. We first give a fast approach to solve the 3-D problem based on the simultaneous evolution of 2-D curves and then give the 3-D formulation using the FEM.

We regard the application of deformable-contour models as a method to extract *smooth shapes* in a given region of the image. The philosophy of the approach is to introduce an elastic curve (or surface) in the image, and let it evolve from an initial position under the action of both internal forces (smoothness constraints, and pressure forces) and external forces (attraction towards local edgels and weight forces).

The paper is organized as follows. After recalling the basic principles of "snakes" (Section II-A) and "balloons," we present enhancements and details about the use of edge data to generate an attraction potential (Section II-B). We also briefly survey the literature. We then define 3-D deformable models (Section II-C) and give a relationship with 3-D edge points (Section II-D). We show two simplified 3-D approaches (Section III) and then finally solve this minimization problem in both 2-D and 3-D by a finite-element method (Section IV).

We illustrate our technique in the application of the automatic segmentation of medical images. The power of the approach to segment 3-D images is demonstrated by a set of experimental results on various complex medical 3-D images (Section V).

II. Energy Minimizing Curves and Surfaces

We first recall some definitions and formulate the mathematical problem. In the following, we will call the active contour model or energy-minimizing curve "the 2-D problem" and the active surface model or energy-minimizing surface "the 3-D problem."

A. 2-D Active Contour Model

1) Definition: Snakes are a special case of deformable models as presented in [29]. The deformable contour model is a mapping:

$$\Omega = [0, 1] \rightarrow \mathbb{R}^2$$
$$s \mapsto v(s) = (x(s), y(s)).$$

We define a deformable model as a space of admissible deformations $\mathcal{A}$ and a functional E. This functional represents the energy of the model which will be minimized and has the following form:

$$E : \mathcal{A} \rightarrow R$$

$$v \mapsto E(v) = \int_\Omega w_1\|v'(s)\|^2 + w_2\|v''(s)\|^2 + P(v(s))ds,$$

where v' and v'' denote derivatives of v and where P is the potential associated to the external forces. The potential is computed as a function of the image data according to the desired goal. If we want the snake to be attracted to edge points, the potential should depend on the gradient of the image. In the following, the space of admissible deformations $\mathcal{A}$ is restricted by the boundary conditions $v(0)$, $v'(0)$, $v(1)$ and $v'(1)$ being given. We can also use periodic curves or other types of boundary conditions.

The mechanical properties of the model are controlled by the functions w_j. Their choice determines the elasticity and rigidity of the model.

If v is a local minimum for E, it satisfies the associated Euler-Lagrange equation:

$$\begin{cases} -(w_1v')' + (w_2v'')'' + \nabla P(v) = 0 \\ v(0), v'(0), v(1) \text{ and } v'(1) \text{ given.} \end{cases} \tag{1}$$

In this formulation, each term appears as a force applied to the curve. A solution can be viewed either as realizing the equilibrium of the forces in the equation or reaching the minimum of the energy.

Thus, the curve is under control of two types of forces

- The internal forces (the first two terms) which impose the regularity of the curve. The constants w_1 and w_2 impose the elasticity and rigidity of the curve.
- The image force (the potential term) pushes the curve to the significant lines which correspond to the desired attributes. It is defined by a potential of the form $\int_0^1 P(v(s))ds$ where

  $$P(v) = -\|\nabla I(v)\|^2.$$

  Here, I denotes the image. The curve is then attracted by the local minima of the potential, which means the local maxima of the gradient, i.e. edges (see [19] for a more complete discussion of the relationship between minimizing the energy and locating contours).

Other forces can be added to impose constraints defined by the user. We will make use of additional forces.

2) Finite-Difference Solution: We first formulate the discretization of the equation by finite differences following [23] in a more succinct fashion. Setting $F(v) = (F_1(v), F_2(v)) = -\nabla P(v) + F_{\text{other}}$, the sum of image and other external forces, the equation

$$-(w_1v')' + (w_2v'')'' = F(v), \tag{2}$$

becomes a linear system after applying finite differences in space:

$$AV = F.$$

Here, A is pentadiagonal and V and F denote the vectors of positions $v_i = v(ih)$ and forces at these points $F(v_i)$ respectively.

Since the energy is not convex, there may be many local minima of E.

Finding the global minimum of the energy does not necessarily have a meaning. Indeed, if v_m is a point of the plane where P has a global minimum, then the constant curve $v(s) = v_m$ is a global minimum for the energy with periodic boundary conditions.

But we are interested in finding a good contour in a given area. We suppose in fact that we have a rough estimate of the curve. We impose the condition to be "close" to this initial data by solving the associated evolution equation

$$\begin{cases} \frac{\partial v}{\partial t} - (w_1v')' + (w_2v'')'' = F(v), \\ v(0,s) = v_0(s), \\ v(t,0) = v_0(0), \quad v(t,1) = v_0(1), \\ v'(t,0) = v_0'(0), \quad v'(t,1) = v_0'(1), \end{cases} \tag{3}$$

where v' denotes differentiation with respect to s. A solution to the static problem (2) is achieved when the solution $v(t)$ stabilizes. This is because the term $\frac{\partial v}{\partial t}$ tends to 0 (generally) and the dynamic system (3) reduces to (2) at infinity.

After formulating the evolution problem using finite differences with time step τ and space step h we obtain a system of the form

$$(\mathcal{I} + \tau A)v^t = (v^{t-1} + \tau F(v^{t-1})), \tag{4}$$

where $\mathcal{I}$ denotes the identity matrix. Thus, we obtain a linear system and we have to solve a pentadiagonal banded symmetric positive system. We compute the solution using a LU decomposition of $(\mathcal{I} + \tau A)$. The decomposition needs be computed only once if the w_i remain constant through time. We stop iterating when the difference between two successive iterations is sufficiently small. After each iteration we test $\|v_t - v_{t-1}\|$ and stop if it is lower than a given threshold. Of course, the lower the threshold, the better we can be sure it is a real equilibrium.

Moreover, the linear system above is such that each row of the matrix $(\mathcal{I} + \tau A)$ is obtained by circularly shifting the previous one. The product of a matrix of this form and a vector can be viewed as the convolution of a row of the matrix with the vector. Since derivatives are at most of the fourth order, this corresponds to a convolution of the discretized curve v_i by a kernel of length five. This smoothing can in fact be viewed as a 1-D low-pass filter on the curve.

Note that in (4), v^t has two components x^t and y^t, and we can write separately the two equations satisfied by the vectors x^t and y^t. These equations are independent except for the term $(F_1(v), F_2(v))$ where x and y cannot be separated. However, as we will see later, in all the iterative schemes we use in this paper, the term $F(v)$ is explicit. This means that at each iteration it may be considered as a constant vector and the two equations satisfied by x and y can be computed separately. Accordingly, we sometimes consider the equation for the unknown v as a scalar function, instead of a two- or three-component vector equation.

The finite difference formulation of the problem makes the curve behave like a set of masses linked by springs of zero length (when fully contracted). Consequently, if there is no image force ($F = 0$), then either the curve shrinks and vanishes to a point, or it straightens out to become a line depending on the boundary conditions.

If the spatial discretization step h along the curve is more than two pixels, the curve can either jump across edges or fail to be attracted to edges. This means that the number of nodes must be of the order of the length of the curve.

The coefficients of elasticity and rigidity have a great effect on the behavior of the evolution of the curve along time iterations. If w_1 and w_2 are close to unity, the internal energy E_{int} has a major influence and the image forces have small effect. In this case the initial curve is merely smoothed due to the regularization action. We are currently studying the effect of these coefficients in simple cases to evaluate the ability of the model to detect corners.

A correct choice for parameters is guided by numerical analysis considerations. We want the coefficients within the rigidity matrix A to have similar orders of magnitude. We obtain good results when the parameters are of the order of h^2 for w_1 and h^4 for w_2, where h is the space discretization step.

B. Improving the Model—The Balloon Model

The potential P is such that the force $F(v) = -\nabla P(v)$ generates the attraction of the curve or surface to the image regions that we seek to extract. Our main goal is the extraction of "good" edge points (i.e., to be able to remove spurious edge points, while insuring connected contours).

The formulation described in the previous section leads to certain difficulties, for which one of us proposed a variation (in [12]) by defining new forces and a potential function. In the following sections, we will extend in a natural way these revised forces for use with the finite-element method for both 2-D curves and 3-D surfaces. In the subsection immediately following, we summarize the main points developed in 2-D in [12], elaborating on certain important details. All these points are identical for a surface evolving in a 3-D image.

1) Normalization of the Force: The external forces based on image data applied to the curve to push it to the high gradient regions are modified to give more stable results. Indeed, it is not possible to choose a uniform time step τ suitable for all points of the contour. If τ is too large, some points on the curve may move too quickly, and jump across the desired minimum and never come back. If τ is too small, very few high gradient points will attract the curve.

So instead of modifying the time step, we modify the force by normalizing it, taking $F = -k\frac{\nabla P}{\|\nabla P\|}$. This simulates a local time step which makes the curve evolve at the same speed everywhere.

2) The Balloon Model. The Weight Force: To make the snake find its way, an initial guess of the contour has to be provided manually. This has many consequences on the evolution of the curve (or surface).

- If the curve is not close enough to an edge, it is not attracted to it.
- If the curve is not subjected to any counterbalancing forces, it tends to shrink on itself.

Accordingly, we introduce an internal pressure by considering our curve as a *balloon* which is inflated. The *pressure* force is added to the internal and external forces to push the curve outward, as if we were introducing air inside. The curve

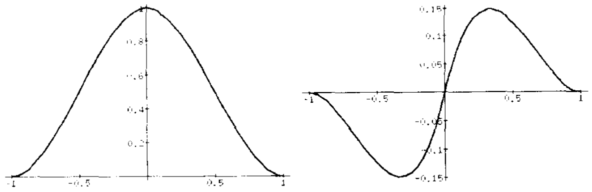


Fig. 2. Plots of the two basis functions ϕ and Ψ for the 2-D FEM.

both expands and is attracted to edges as before. But if the edge is too small or too weak with respect to the pressure force, the curve passes over the edge, growing outward.

The internal pressure force prevents the curve from being "trapped" by spurious isolated edge points, and makes the final result much less sensitive to the initial conditions.

The force F now becomes

$$F = k_1\vec{n}(s) - k\frac{\nabla P}{\|\nabla P\|}(v(s)), \tag{5}$$

where $\vec{n}(s)$ is the unit vector normal to the curve at point $v(s)$ and k_1 is the amplitude of this force. The coefficients k_1 and k are chosen such that they are of the same order, which is smaller than a pixel size (the length unit), with k slightly larger than k_1, so an edge point can stop the inflation force.

Remark that this force can also be interpreted as the gradient of an extra energy term. This would be a surface term $E_{\text{area}} = -k_1\int dA$, measuring the area inside the region delimited by the curve. Minimizing this energy corresponds to have the inside region as large as possible, which is obtained by a force pushing in the direction of the external normal.

Note that F depends on not only the position $v(s)$, but also of the normal at this position. In the iterative methods presented in this paper, we solve problems formulated under the assumption that F depends on the position v, but not on derivatives. This assumption is made possible by using as an approximation to v^t at step t the previously computed value v^{t-1}.

Suppose we have an image of a black rectangle on a white background, and a curve is placed inside the rectangle. Without the inflation force, even if we have perfect edge detection, the curve will shrink and vanish. Starting from the same small curve, but using the inflation force, we obtain the entire rectangle (see Fig. 4). When the balloon reaches equilibrium, the points that are attracted to image edges are slightly outside of the real contour. We thus reduce the inflation force to localize the final position of the curve.

As another example, we apply the technique to a slice from a 3-D image of the region of the heart obtained with magnetic resonance imaging (MRI). We wish to extract the left ventricle. We use here the 3-D edge detector [24] obtained by generalization of the 2-D Canny–Deriche filter. In Fig. 7, we show the result of the application of balloons to detect the ventricle. The initial curve was neither close in shape nor in position to the actual ventricle.

One aspect of the increased complexity of the method is a large variation of the length of the curve between the initial data and the final limit curve. As we remarked above, the

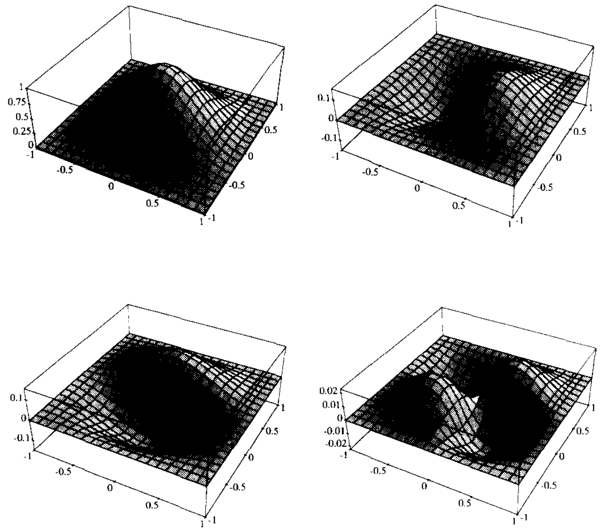


Fig. 3. Surface plots of the four basis functions φ, ψ, η and ζ for the 3-D FEM.

Fig. 4. Advantage of the balloon model: the initial curve (in black, on the left) neither collapses nor gets trapped by spurious isolated edge points. It robustly converges toward the desired rectangle shape (on the right). The background image is the attraction potential generated by hand-drawn contours (see Fig. 5).

number of nodes along the curve should be approximately equal to the length of the curve. Thus we must change the discretization during the iteration process. To do this, we periodically reparametrize the curve, and resample node points. This means that we construct a new parametrization using the existing curve by sampling at a one pixel distance between nodes. This also prevents nodes from clustering at high gradient points and from separating, creating a large space between some nodes.

Since the length changes, we must change the matrix A during the iteration process. Accordingly, our algorithm incorporating internal pressure takes more time to converge, since we must compute matrix inverses at each reparametrization and also since we begin with a curve very far from the solution. The added computation time is a price we must pay for the simplicity of specifying a coarse initial curve.

In the same spirit as the balloon model, we will also incorporate a "*weight force*" into the 3-D reconstruction models. The weight force allows us to take a very simple initial surface placed on the border of the image. The surface then "falls" under the influence of the "gravity," to catch an object which might be far from the border. If we instead attempt to locate

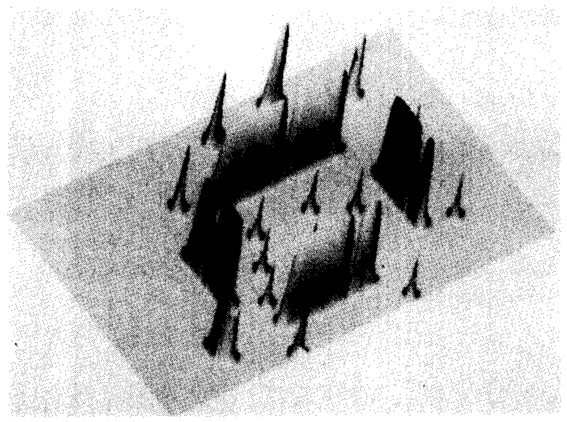


Fig. 5. Attraction potential surface generated by convolution of a Gaussian and the edge contours defined by hand shown in Fig. 4. The surface is shown upside down for sake of clarity. The curve is attracted to minima of the potential, which are maxima as seen in the figure. The potential around isolated points shows the shape of theGaussian used. The attraction force is small outside a neighborhood of an edge point.

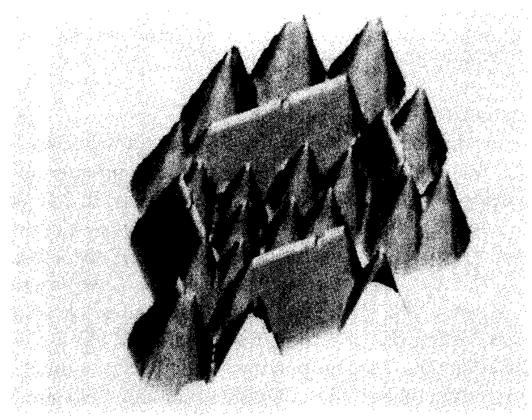


Fig. 6. Surface of distances to the nearest edgepoint. The negative of this surface, as the previous one, may be used as a potential.

a surface by surrounding the outside of the object by the deformable model and then use a "deflation" force (identical to the inflation force, but with a negative k_1), instability can result since the surface may then self-intersect after a few iterations.

The "weight force" is uniform on the surface in direction and intensity: $F = k_1 \vec{Z}$. The initial surface is typically a plane on one side of the 3-D image, and $\vec{Z}$ is defined to be normal to this plane. As with the inflation force, if the weight force is not turned off at the end of the process, equilibrium is reached with the surface slightly shifted from the desired solution. In the weight force case, however, we eliminate the force locally at a point when an area of large variation of the gradient is reached instead of once global convergence is obtained for the pressure force. This modification improves the progression to the solution. As a result, we may use larger values of k_1 and thus move faster without missing the solution. As with the balloon model (see [12]), the surface is not stopped by isolated spurious points. The effect of the weight force will be

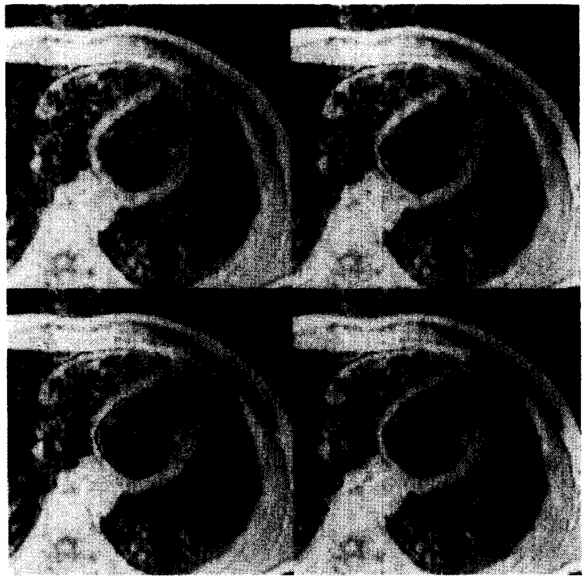


Fig. 7. An MRI image. Evolution of the balloon curve to detect the left ventricle. Here, we give illustration of the robustness of the balloon model: The final result can be achieved from almost any initial curve given within the interior of the ventricle (see also Fig. 11).

demonstrated in Figs. 15 and 16.

3) Accounting for Prior Local Edge Detection:Attraction Potential: We make use of *edge points* extracted prior to the use of the deformable model by a local edge detector. In 2-D, edge points tend to lie along curves, and in 3-D they are located on surfaces. Accordingly, we are able to combine the qualities of a good *local* edge detector, such as the Canny-Deriche edge extractor [7], [17], [24], with a *global* active model. We must define the attraction forces through the use of a potential function. The potential may be defined by convolving the binary edge image with a Gaussian impulse response. An example is shown in Fig. 5, plotting the potential surface generated by the rectangle image of the previous section (Fig. 4).

We also used in [13] a Chamfer distance that approximates the Euclidian distance to the nearest edgels [5], or a Euclidean distance image (as defined in [15]). Fig. 6 shows a potential based on the latter distance map for the same rectangle image as before. These approximate distance metrics are of interest because they can be obtained by a fast algorithm, requiring only two-passes through the binary image.

We denote by $d(v)$ the distance between a point v and the nearest edge. In general, a large class of potentials may be formulated as $P(v) = g(d(v))$, i.e., as a function of the distance to the closest contour. For instance,

$$P(v) = -e^{-d(v)^2}$$

produces a potential that is similar to the Gaussian convolution method discussed above, except that only the closest edge point has an effect at a position v. The potential

$$P(v) = \frac{-1}{d(v)}, \quad (P \equiv -1 \text{ if } d(v(s,r)) < 1),$$

where the unit distance is the pixel size, produces a faster convergence since this potential decays more slowly, producing larger forces at points distant from the edges.

Remark that if the potential is defined by $P(v) = g(d(v))$, the force becomes $F(v) = -\nabla P(v) = -g'(d(v))\nabla d(v)$. When this force is normalized as suggested in Section II-B-1, we have $F = -k\frac{\nabla d}{\|\nabla d\|}$. The formula does not depend on function g but the numerical result may be different because of machine accuracy. So, when we normalize the force, we could take any function g easy to compute, for example $g(d) = d$, but the distance function is not differentiable everywhere. This is why g usually behaves like d^2 for small d to avoid problems at points where $d = 0$. In general, g is also used to regularize the distance function d.

However, in the case of a potential defined from a distance function, it may be better when the force is not normalized and the norm of F depends on $g'(d)$ and $\|\nabla d(v)\|$. Using the triangular inequality, we can see that $\|\nabla d(v)\| \leq 1$ (this is in fact equal a.e.). So, a good choice of g permits to control the norm of the attraction force when d is small or large. This will be discussed in the following.

The attraction forces derived from the potential may be used either as the only image forces, or may be combined with an intensity-gradient image to enhance the detected edges. The latter approach is useful when the detected edges are broken into small disconnected segments.

The methods of convolving edges with a Gaussian and defining a function of $d(v)$ were used by us in [11]–[13]. However, the attraction potential defined by Weiss [34], and the weak-continuity method of Blake and Zisserman [4], are closely related. The attraction potential and weak-continuity methods are applied to sparse isolated points, our set of edge points are extracted by a local edge detector, and thus may contain full curves (or surfaces in 3-D). Moreover, in these methods, the model tries to match the whole data, while we are doing segmentation at the same time. Our deformable model has to find out which parts of the data to stick to. Also, the goals are different. The property of the varying mesh model is to define automatically an optimal mesh to deal better with corner reconstruction, using an extra potential term. In our model, the inflation force is a powerful tool to make the model converge to the solution being less demanding of the initialization. These two tools could be associated together to obtain both properties.

In the next section we survey more closely the definition of attraction potential in the reconstruction literature.

4) A Survey of Attraction Potential Used in Reconstruction Methods: The general formulation of the problem as presented in [34] uses Tikhonov regularization [33] to approximate data g by a smooth function f, in order to reconstruct a curve or a surface. We use a second-order regularization scheme to insure a C^1 continuity of the solution. Two terms are minimized:

- a criteria of the faithfulness to the data; and
- a regularizing term containing derivatives of the function.

The energy functional can be written in the form:

$$E(f,g) = \int V(f(s), g(s))ds + \int S(f(s))ds, \quad (6)$$

where V is a measure of the distance between the function f and the data g, and $S(f)$ measures the smoothness of the reconstruction f. Similar to our potential P, the attraction force is obtained from the gradient of V, $F_V = -\nabla V$.

Let us consider the case of a curve $f(s) = (x(s), y(s))$ and discuss the different approaches to reconstruction. We also give an interpretation of the forces by means of zero-length spring attraction forces. All of our discussion generalizes naturally to surfaces in 3-D data.

Least-Squares—Explicit Constraints: The most classic problem is least-square fitting given the position of the curve at a collection of points $f_i = f(s_i) = (x_i, y_i)$ at known values of the parameter s_i. We use as an attraction potential

$$V(f) = \sum_i \|f(s_i) - f_i\|^2,$$

(see, e.g., [28], [25] and their references). The case of a *cartesian curve* is especially simple since $x_i = s_i = x(s_i), x(s) = s$ and $V(f) = \sum_i (y(s_i) - y_i)^2$; this is the case treated in [25].

The attraction force obtained by differentiation of the potential is proportional to the distance between a data point f_i and the value of f at s_i. We can interpret this force physically as a spring (which contracts to zero length) connecting a point of the curve (or surface) $f(s_i)$ and the given point $f_i = (x_i, y_i)$. Each node of the curve is connected by a spring to one *explicit* data point. Thus each data point (x_i, y_i) influences the force at only one point of the curve. The curve is constrained to best fit to *all* the data. Moreover, the data points must be sorted in a natural order. This is the case for a cartesian curve (or surface), where values s_i correspond to positions along an axis (two axes for a surface). For a general curve, given a collection of (x_i, y_i) data, the natural order of the points may not be so apparent.

Position-Independent—Implicit Constraints: When the s_i are not given and the curve has to best fit the set of points f_i, a simple extension of the previous idea would define an attraction potential simulating a zero-length spring for each data point of the plane which has effect for any point of the curve. At a point h of the plane, the potential is the sum of the contributions of all the f_i: $V(h) = \sum_i \|h - f_i\|^2$. The potential V thus may be viewed as a convolution of the sum of Dirac masses δ_{f_i} at the data points f_i with the function $\|h\|^2$:

$$V(h) = \int \sum_i \delta_{f_i}(u)\|h - u\|^2 du = \sum_i \|h - f_i\|^2.$$

This potential has the advantage of being convex, but does not work out well since a point of the curve will be attracted with the strongest force by the most remote data point. Indeed the only minimum is a curve reduced to a point located at the mean value of the f_i's.

Our approach convolves a binary image of edge points with a function of the form $-e^{-\|h\|^2}$, while Weiss in [34] convolves a set of sparse data points with a similar function of the form $\frac{-ae^{-\|h\|^2/b^2}}{1+c\|h\|}$. We chose a negative Gaussian function since it behaves like $C + \|h\|^2$ for small h (where C is a constant) and has a zero limit at infinity. Thus, the attraction force behaves like a zero-length spring when h is small, and when h becomes large the force decreases to zero. So the curve is most attracted by the points close to it, and distant points have no attraction force.

Blake and Zisserman's functions $g_{\alpha,\lambda}$ and $g^*_{\alpha,\lambda}$ of [4, fig. 7.1] are likewise similar in structure. However, their forces are used to define the internal attraction between two successive points in a discretized curve. The idea of the weak continuity is that if the variation is too large at some point of the curve, then it is better to break the reconstruction curve there and introduce a discontinuity. The weak continuity makes springs defining the internal attraction force (see Section II-A-2) break if they get too long. The attraction force based on the image data that we use here is thus similar to the internal attraction force in the weak continuity model of [4].

Note that the attraction force for the convolution-based potential allows the curve (or surface) to choose among the data points the ones to fit. Each point of the curve is attracted by all the data *points close enough* to it.

When we define the potential by $P(v) = -e^{-d(v)^2}$ using the Chamfer distance [13], each point is attracted only by the closest edge point. The curve behaves as though each point is linked by a weak spring (which breaks if too long) to the *closest data point.* The constraints in this part are not known explicitly like in the "classic" reconstruction but defined *implicitly* by the relative position of a node to the data.

Snakes: In some cases, the data is not known explicitly. For example, the potential introduced with the snakes in [23] is based on the property of edge points to have a large image-gradient value. The potential defined as a function of the image-gradient results in the curve being attracted to the high gradient points without explicit knowledge of these points. The constraints are also *implicit* in this case. In the snake approach, the data points are located directly by the curve through the minimization of the potential (see Section II-A). Moreover, all the points of the curve are influenced by the attraction forces from the image.

Mixed Version: Recently a combination of the previous approaches was proposed in [16]. Two potentials are defined. A "data energy" term is used to represent an attraction of the surface to the closest data point, which yields a force that is linear when close to the data and decreases to zero when far from the point. The data energy is the same as our potential using the Chamfer distance. A "feature energy" term represents an attraction to feature points. Though the function of convolution is slightly different in form, it has similar properties to the "weak spring" model discussed above. The main difference is that this "feature potential" is modified with iterations. The threshold at which the spring breaks decreases from a reference distance when $t = 0$ to zero when $t = T_0$. Therefore, the influence of features decreases during the evolution.

In this section, we presented the enhancements of our model, normalizing the force to get more stability, adding an inflation or weight force to push the model more quickly to the solution, and defining an attraction potential making use of edge points extracted prior to the application of the model. For this last point, we gave a survey of the related attraction potential found in the literature.

We will make use, in this paper, of all the features presented so far for 2-D curves as well as 3-D surfaces.

C. 3-D Active Surface Model

The 3-D model is obtained by generalizing the formulations given in the previous sections. A surface $\mathcal{S}$ is defined by a mapping v:

$$v : \Omega = [0,1] \times [0,1] \to \mathbb{R}^3$$
$$(s,r) \mapsto v(s,r) = (v_1(s,r), v_2(s,r), v_3(s,r))$$

and the associated energy E is defined on an admissible class $\mathcal{A}$ of mappings v, and has form:

$$E : \mathcal{A} \to \mathbb{R}$$
$$v \mapsto E(v) = \int_\Omega w_{10} \left\| \frac{\partial v}{\partial s} \right\|^2 + w_{01} \left\| \frac{\partial v}{\partial r} \right\|^2 + 2w_{11} \left\| \frac{\partial^2 v}{\partial s \partial r} \right\|^2 + w_{20} \left\| \frac{\partial^2 v}{\partial s^2} \right\|^2 + w_{02} \left\| \frac{\partial^2 v}{\partial r^2} \right\|^2 + P(v(s,r)) ds\, dr, \quad (7)$$

where $P(v(s,r)) = -\|\nabla I(v(s,r))\|^2$ is the potential associated with the external forces. The internal forces acting on the shape of the surface depend on the coefficients w_{ij} such that the elasticity is determined by (w_{10}, w_{01}), the rigidity by (w_{20}, w_{02}), and the resistance to twist by (w_{11}). That is, the coefficients determine the mechanical properties of the surface. We can also constrain the surface structure by adjusting boundary conditions (for instance, to create a cylinder or a torus). This model, restricted to its first-order derivative terms, may be interpreted physically as a membrane, and with inclusion of second-order derivative terms may be interpreted as a thin plate.

A local minimum v of E satisfies the associated Euler-Lagrange equation:

$$\begin{cases} -\frac{\partial}{\partial s}\left(w_{10}\frac{\partial v}{\partial s}\right) - \frac{\partial}{\partial r}\left(w_{01}\frac{\partial v}{\partial r}\right) + 2\frac{\partial^2}{\partial s \partial r}\left(w_{11}\frac{\partial^2 v}{\partial s \partial r}\right) \\ + \frac{\partial^2}{\partial s^2}\left(w_{20}\frac{\partial^2 v}{\partial s^2}\right) + \frac{\partial^2}{\partial r^2}\left(w_{02}\frac{\partial^2 v}{\partial r^2}\right) = F(v) \end{cases} \quad (8)$$

subject to boundary conditions. The Euler-Lagrange equation is a necessary condition for a minimum. As with (5), F denotes the sum of forces: $F = F_{\text{image}} + F_{\text{balloon}}$, F_{image} is the force obtained after normalization from the gradient of the attraction potential, and F_{balloon} is either the inflation or weight force. Since the energy function is not convex, there may be many local minima of E. The Euler–Lagrange equation (8) is satisfied at any such local minimum. But as we are interested in finding a 3-D contour in a given area, we assume in fact that we have a rough *prior* estimation of the surface. This estimate is used as initial data for the associated evolution equation, in which we add a temporal parameter t:

$$\begin{cases} \frac{\partial v}{\partial t} - \frac{\partial}{\partial s}\left(w_{10}\frac{\partial v}{\partial s}\right) - \frac{\partial}{\partial r}\left(w_{01}\frac{\partial v}{\partial r}\right) + 2\frac{\partial^2}{\partial s \partial r}\left(w_{11}\frac{\partial^2 v}{\partial s \partial r}\right) \\ + \frac{\partial^2}{\partial s^2}\left(w_{20}\frac{\partial^2 v}{\partial s^2}\right) + \frac{\partial^2}{\partial r^2}\left(w_{02}\frac{\partial^2 v}{\partial r^2}\right) = F(v), \\ v(0,s,r) = v_0(s,r) \text{ (initial estimate)}, \end{cases} \quad (9)$$

and where boundary conditions may have to be imposed (additionally). A solution to the static problem is found when the solution $v(t,s,r)$ converges as t tends to infinity. Assuming sufficient uniform convergence is achieved, the term $\frac{\partial v}{\partial t}$ vanishes, thus providing a solution of the static problem.

With this formulation, and with the potential as given above, the resulting surface will accurately locate the 3-D edge points.

Before describing the numerical solutions of the 3-D reconstruction we give in the next section a mathematical result showing how the surface locates on the Canny's 3-D edge surfaces.

D. Minimizing Surfaces and 3-D Image Edge Points

We comment on the relationship between the surface minimizing the energy of external forces E_{image} and 3-D edge points. A similar formulation for planar curves is given by Fua and Leclerc [19]. Recall that the external energy is given by

$$E_{\text{image}} = \int\int P(v(s,r)) ds dr.$$

We use the following definition of the 3-D edges, as proposed by Canny [7].

Definition 1: A 3-D edge is a surface $\mathcal{S}$ whose points have a minimal potential in the direction normal to the surface. All points along the surface $\mathcal{S}$ satisfy:

$$D_{N(v(s,r))} P(v(s,r)) = 0, \quad (10)$$

where $N(v(s,r))$ is the normal to the surface $\mathcal{S}$ parametrized by $v(s,r)$, D_N is the directional derivative in the direction N, and P is the potential to be minimized.

When the potential is defined in terms of the image gradient ∇I (where typically, I is replaced with a Gaussian-convolved version of the image), the former definition is the same as Canny's edge points.

Definition 2: A 3-D Canny edge is a surface $\mathcal{S}$ whose points have a maximal gradient magnitude in the direction normal to the surface. All points along the surface $\mathcal{S}$ (called Canny's edge points) satisfy:

$$D_{N(v(s,r))} \|\nabla I(v(s,r))\| = 0 \quad (11)$$

where ∇I is the gradient magnitude.

To explore the relation between the energy minimizing surfaces and this definition, let us define the energy associated to the external forces as

$$E_P(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \int\int P(v(s,r)) dA, \quad (12)$$

where $|\mathcal{S}| = \int\int \|v_s \wedge v_r\| ds dr$ is the surface area and $dA = \sqrt{EG - F^2} ds dr$ is the standard surface area measure.

In [14], we show that a surface $\mathcal{S}$ is a local minimum of E_P, with respect to infinitesimal deformation, if:

$$D_{N(v(s,r))} P(v(s,r)) = \frac{eG - 2fF + gE}{EG - F^2} \left(P(v(s,r)) - \frac{1}{|\mathcal{S}|} \int P(v(s,r)) dA \right), \quad (13)$$

where $E(s,r)$, $G(s,r)$, $F(s,r)$, $e(s,r)$, $f(s,r)$ and $g(s,r)$ are the coefficients of the first and second fundamental forms in the basis $\{x_s, x_r, N\}$ (using the same notation as in [18]). A remarkable result is that the quotient $\frac{1}{2}\frac{eG-2fF+gE}{EG-F^2}$ is simply the mean curvature of the surface $\mathcal{S}$.

Equation (13) shows that there exists two interesting special cases:

1) if a minimizer of E_P is a minimal surface (i.e., a surface with a mean-curvature which is everywhere zero), then it is automatically a 3-D edge;
2) if the minimizing surface is composed of edgels with constant Potential, then the term within parentheses in (13) vanishes, and the surface is again a 3-D edge.

In general, these are interesting but exceptional academic situations, and the deformable model simply converges to a solution which is a balance between the applied external forces (corresponding to the energy E_P) and the internal forces, parametrized by the elasticity coefficients w_{ij}. The directional derivative will satisfy (13), but not in general be zero. But in practical implementation, this is approximately the case when the surface is smooth or when the potential has small variation along the surface.

III. Simplified 3-D Model

The main difficulty in passing from modeling curves in 2-D to modeling surfaces in 3-D is the very significant growth of the computation time due to the size of the system to solve. In this section, we describe a much-simplified 3-D surface model, with the aim of minimizing the computational requirements.

A 3-D image is viewed as a sequence of 2-D images which we call slices or cross sections. In this section, we first present a 3-D reconstruction method based on successive solutions of 2-D problems, then show how the 3-D deformable model may be simplified to a simultaneous solution of 2-D problems interacting to yield a fast algorithm.

A. 3-D Reconstruction from a Sequence of 2-D Contour Models

In [12], we reported initial experiments with 3-D reconstruction using a method that directly extends the 2-D method. In this work, we extracted the contour slice by slice. For each slice, a 2-D model is applied. In order to improve the speed of the algorithm, the result of the previous slice is used for solving the successive slices. Assuming that the variations are small from one slice to the next, this works well, in the same way that snakes are used for temporal tracking in [22], [23].

In order to reconstruct the entire 3-D surface, we initialize the process using a curve obtained from the balloon model in an intermediate cross section, and then propagate the result to neighboring cross-sections. In [1], a related approach was taken, but successive curves were extracted by hand from each slice, using an edge image from each slice. Note that the inflation force is necessary only for the first slice, to have a good solution on that slice, beginning with a bad initial data. But in the following slices, the inflation force is not used since the solution of the previous slice is already close to the solution of the current slice.

Fig. 8 shows a reconstruction of the left and right ventricles using data from a 3-D magnetic resonance image of the heart region. This reconstruction is nearly automatic, although when the contour undergoes a large change from one slice to the next, the initial curve in that slice may have to be redefined in order to obtain a good contour. This problem can be ameliorated by adding interpolation slices when necessary. We note that the problem never occurs in practice when the image resolution is the same in the three axes.

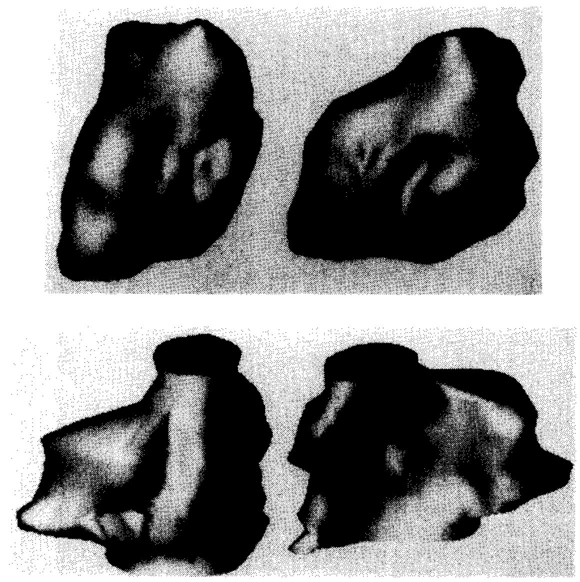


Fig. 8. 3-D reconstruction from a sequence of 2-D contour models: two views of the reconstructed inside cavity of the left and right ventricles.

The entire 3-D surface represented as a sequence of contours of the slices. We use the NUAGES software package (see [1]) to display the results, which also has the capability to define a 3-D surface from planar curves.

The main issues of that approach are that first, there is no interaction between slices and second, the surface has to be cylindrical. The first point is solved in next section while the second will need the general model of Section IV.

B. Fast Solution of the 3-D Constrained Problem

We next describe a fast approach to solve the 3-D problem based on the simultaneous evolution of 2-D curves.

The 3-D deformable model is obtained by minimizing the energy term (7) defined in Section II-C which uses parametrized surfaces $v(x) = v(s,r) = (v_1(s,r), v_2(s,r), v_3(s,r))$.

In order to keep the model simple, we limit degrees of freedom of the deformation to two components instead of three by constraining the third component v_3, which corresponds to the slice level, to depend only on r. In this case, the third component of the external force is zero.

Thus, the surface that we seek is represented as a sequence of planar curves, with the second parameter r being the index of the slice. We have $v(s,r) = (v_1(s,r), v_2(s,r), r)$ so that for each fixed value of the parameter r, there corresponds a closed curve parametrized by s lying in a slice of the 3-D image.

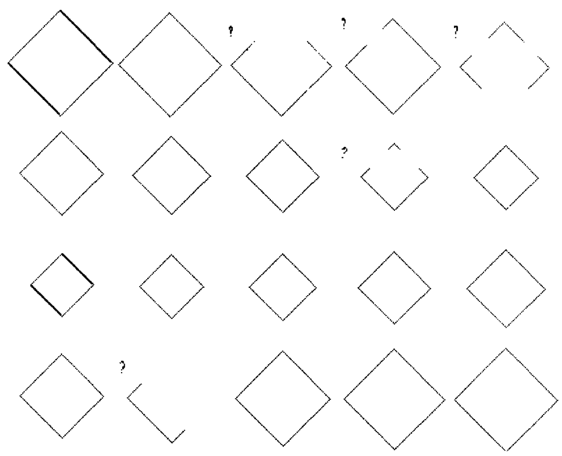


Fig. 9. Edge image of the frustum after erasing some parts. The cross sections with a "?" were modified. The sequence of cross sections is ordered from left to right and from top to bottom. The square shrinks to an intermediate size, and then increases back to the original size.

The consequences of constraining the surface as a sequence of plane curves entails two advantageous simplifications.

- First, the curves of the representation are necessarily separated, and undesirable deformations which would require a new parametrization of the surface are avoided. Although this imposes a restriction on the surfaces that we can reconstruct, the representation involves distinct curves, one per slice. As a result, if a contour is missing in a slice, the surface nonetheless bridges the neighboring contours, creating a smooth surface.
- Second, the extraction of information within a slice is simplified, both during the iterative construction as well as for the final result, since the surface is represented by slices. In the more general case, where nodes can move between slices, it is nontrivial to compute the contours resulting from the intersection of the surface and a slice. In the approach we present at the end of the paper for the general 3-D model, we in fact make use of such a computation for visualization of information on slices.

The main difference between the slice-by-slice approach of the previous section and the constrained 3-D approach of this section is first, that an interaction is permitted between neighboring slices and also their simultaneous evolution. If edges are missing in a particular slice, then the previous method will fail, whereas using $v(s,r)$ opens the possibility that edges missing from a sequence of slices will be filled in. We can see in Figs. 9 and 10 how such missing edges are retrieved. This is illustrated by the bottom middle slice in Fig. 10 where a 2-D deformable model would not reconstruct the missing edges. It would only smooth the data and give a small rectangle corresponding to the available edges. With the simplified 3-D model, the curve on one slice is also attracted by the edge data of the neighboring slices, and this helps to reconstruct the whole curve (a square in the example).

We recall that the solution is obtained by minimizing the energy of (7). A minimum v of the energy satisfies the Euler

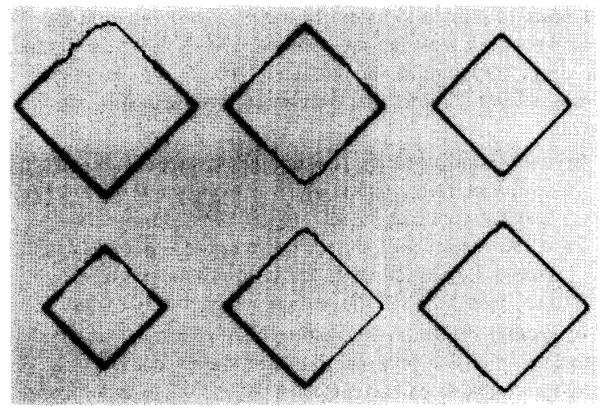


Fig. 10. The simplified 3-D model: results for the frustum on six slices. They are ordered from left to right and from top to bottom. The two cross sections on the far right correspond to complete contours, the other ones had some parts erased.

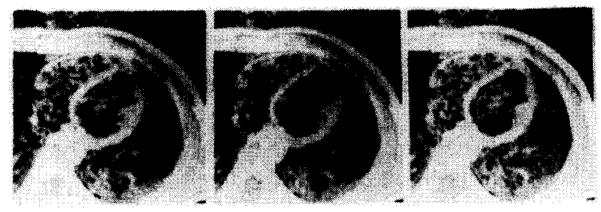


Fig. 11. Using the FEM. Evolution (from left to right) of the balloon curve to detect the left ventricle (see also Fig. 12).

equation (8), and a solution to the static problem is found when the solution $v(t)$ of nine stabilizes. In fact, since $v_3(s,r) = r$, only the first two components of Eqns. 8 and 9 are nonzero. Solving this equation with finite differences, we obtain a 2-D linear system of the form:

$$\frac{V^t - V^{t-1}}{\tau} + AV^t = F(V^t), \tag{14}$$

where V^t is the vector whose components are the values of $(v_1(s,r), v_2(s,r))$ at the nodes of the discretization at iteration t. The first unknown vector V^0 is given by the initial data. Assuming V^{t-1} calculated, we solve Eqn. 14 with respect to V^t.

The unknown V^t appears in the three terms of (14). We say that the scheme is *totally implicit*. It is difficult to solve since the force F has a complicated form. We can approximate V^t by V^{t-1} in the term $F(V^t)$, like in the 2-D case (4). We then say that we solve a *semi-implicit* scheme, i.e., one that is explicit in the force term (F) and implicit for the matrix term A. This will also be the case for the finite-element method, discussed in the following sections. Inversion of the 3-D scheme of this section is obtained by solving the system

$$V^t = (\mathcal{I} + \tau A)^{-1}(V^{t-1} + \tau F(V^{t-1})).$$

It is easier, and thus faster, to solve the *totally explicit* scheme where the term AV^t is also approached by the known value AV^{t-1}. The unknown V^t is then directly calculated without matrix inversion by the formula:

$$V^t = (\mathcal{I} - \tau A)V^{t-1} + \tau F(V^{t-1}).$$

Note that this explicit scheme is a first order development of $(\mathcal{I} + \tau A)^{-1}$ as $(\mathcal{I} - \tau A)$. In practice, both $(\mathcal{I} + \tau A)^{-1}$ and $(\mathcal{I} - \tau A)$ perform a smoothing operation on the data V^{t-1}. So, in our implementation, we first add the forces at each iteration, and then smooth v to remove singularities. This does not change the global behavior and gives, at each iteration, a better estimation for visualization of the intermediate results.

There is a certain anisotropy introduced by the restriction on the third axis direction, but there is some justification for the choice, since the data itself possesses this structure.

Although we solve a 3-D problem, the discretized surface can be represented by the set of two two-dimensional arrays $v_1(s_i, r_j)$ and $v_2(s_i, r_j)$). Since our scheme is explicit in the term $F(v)$, we can consider separately the two components v_1 and v_2 at each iteration as noted in Section II-A-2. We now present the steps of the algorithm.

1) *Addition of the expansion force:* Each boundary node along the curve at each slice is moved along vector $k_1 \overrightarrow{n}(s, r)$ where $\overrightarrow{n}(s, r)$ is the external normal to the planar curve of the slice at level r.
2) *Addition of the edge force:* At each node, we add $-k\frac{\nabla P}{\|\nabla P\|}$, where the third component of the gradient is arbitrarily set to zero to constrain the force to lie within the slice plane. The potential P is obtained from the gradient computation $\|\nabla I\|$ of the 3-D image data.
3) *Smoothing:* We apply a smoothing operation by means of the matrix $(\mathcal{I} - \tau A)$ to the data V^t, separately to its components v_1^t and v_2^t. It can be verified that this matrix is banded and that the circular shifting property of the matrix in the 2-D problem (see Section II-A-2) is extended to our 3-D problem. The product of a matrix of this form and the discretized surface can be viewed as the convolution of a two dimensional array with the arrays $v_1(s_i, r_j)$ and $v_2(s_i, r_j)$. Since derivatives are at most of the 4th order, this corresponds to a convolution of the discretized surface $v(ih, jk)$ by a 5×5 kernel. This smoothing can in fact be viewed as a low-pass filter. The strength of the filter is determined by the coefficients of matrix A and the time step τ.

The third step can be made faster by decomposing the convolution of the surface nodes by the 5×5 kernel into a product of two 1-D low-pass filters of length 5, one in each direction, approximating the convolution kernel by a separable kernel. Although the kernel $(\mathcal{I} - \tau A)$ is not precisely a separable kernel, due to the term $w_{11}(x)\left\|\frac{\partial^2 v}{\partial r \partial s}\right\|^2$, the following procedure provides an acceptable approximation.

- *Smoothing in a cross section:* Recall that in $v_1(s, r)$ and $v_2(s, r)$, the r variable gives the slice level and s gives a spatial location along the curve. For each slice separately, the smoothing operation restricted to this slice is performed. That is, for a given slice r, the terms of the matrix $(\mathcal{I} - \tau A)$ involving data on slice r are performed. This is a first order approximation of the smoothing in the 2-D problem.
- *Smoothing between cross sections:* Smoothing in the direction orthogonal to the slice planes.

Figs. 9 and 10 show example applications using a simple shape. For 20 slices of a 3-D image, a white square on a black background is placed in the center of the image, to form a *frustum*: from slice to slice, the size of the square decreases and then increases again.

The 3-D edge image is used to define a potential P as described in Section II-B-3. The initial curves needed to start the process in the successive cross sections form a cylinder with a square cross-section centered on each slice whose size is smaller than that of the desired objects.

However, for the examples just described, the stack of 2-D models would work just as well to find the solution. Therefore we delete from the edge image large parts of the edges in many of the cross sections (see Fig. 9). If a 2-D model were applied slice-by-slice in this case, the method would close the contours, but inaccurately track the shape. Here, the 3-D smoothing step restores the missing parts in each slice in a coherent way from the edges in the neighboring slices, as shown in Fig. 10.

This method especially fast if the initial approximating surface is a thin tube inside the region of interest.

IV. Numerical Solution by Finite-Element Method (FEM)

The main problem with the 3-D model is the very large number of variables, and the concomitant computation time. In the previous section, we described a simplified model, which gives satisfactory results in some cases, but generally requires further refinement. The simplified approach is only useful for tubular shapes about a single axis. To search for more general surfaces, we must solve a more complete problem, without shape constraints, except for a topological constraint defined by the boundary conditions.

As a remedy to the computational costs, we use the finite-element method (FEM), which is able to effectively lower the number of discretization nodes. Initial experiments comparing a FEM and FDM in the 2-D problem suggests that the FEM has a lower complexity, which becomes more important in 3-D since a greater number of discretization nodes are required as compared to the 2-D problem (see Section IV-C). Thus, in this section, we first present the FEM formulation of the problem in 2-D, then its generalization to 3-D, and then details can be found in [14].

A. Mathematical Formulation

We consider the evolution equation of (3):

$$\begin{cases} \frac{\partial v}{\partial t} - \frac{\partial}{\partial s}\left(w_1 \frac{\partial v}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(w_2 \frac{\partial^2 v}{\partial s^2}\right) = F(v) \\ v(s, 0) = v_0(s) \text{ initial curve} \\ + \text{ Boundary conditions.} \end{cases} \tag{15}$$

In 3-D, the equation becomes:

$$\begin{cases} \frac{\partial v}{\partial t} - \frac{\partial}{\partial s}\left(w_{10}\frac{\partial v}{\partial s}\right) - \frac{\partial}{\partial r}\left(w_{01}\frac{\partial v}{\partial r}\right) + 2\frac{\partial^2}{\partial s \partial r}\left(w_{11}\frac{\partial^2 v}{\partial s \partial r}\right) \\ + \frac{\partial^2}{\partial s^2}\left(w_{20}\frac{\partial^2 v}{\partial s^2}\right) + \frac{\partial^2}{\partial r^2}\left(w_{02}\frac{\partial^2 v}{\partial r^2}\right) = F(v) \\ v(0,s,r) = v_0(s,r) \text{ initial estimation} \\ + \text{ Boundary conditions.} \end{cases} \tag{16}$$

In this case, $v = (v_1, v_2, v_3)$, and is a function of one time variable and of two spatial variables. To simplify the notation, we will consider Eqns. 15 and 16 only with zero-boundary conditions. More general cases can be handled by a simple change of variables. Moreover, as we noted in Section II-A-2, each component of v will satisfy the same equation and may be computed separately. We thus limit in the following to the resolution of the FEM for a scalar-valued function.

In the following subsections, we describe the different steps which lead to the numerical solution of the partial differential equation characterizing the deformable models.

1) Variational Problem: An approach for solving the above equations is to define the associated variational problem. The main idea can be understood by saying that the terms of (16) are equal in some functional space if their scalar product against any vector of the space are equal. This variational problem characterizes the solution of the partial differential equations by defining the space of admissible solutions and its norm using a bilinear form $a(\cdot,\cdot)$ (characterizing the space norm) and a linear form $L(\cdot)$ (characterizing the input).

In [14], we show how to define properly the bilinear form $a(\cdot,\cdot)$ and the linear form $L(.)$ such that solving Eqn. (1) or more generally (8) is equivalent to solve the variational problem:

Variational Problem 1: Find $v \in H_0^2(\Omega)$ such that:

$$a(v,u) = L(u) \quad \forall u \in H_0^2(\Omega), \tag{17}$$

where the space $H_0^2(\Omega)$ is the Sobolev space of functions v such that $\int |D^m v|^2 < +\infty$ for $m = 0,1,2$ where $D^m v$ is the m^{th} order differential of function v.

The existence and uniqueness of a solution to this variational problem (17) are easily established [8], since the bilinear form $a(.,.)$ is symmetric and coercive providing $w_{kl}(s,r) > 0$.

Here, L is assumed to be independent of v. In fact, L does depend on v in our application, but in the iterative scheme, we will fix L to be constant in any given iteration (see [20] for a mathematical justification).

We give expressions for $a(\cdot,\cdot)$ and $L(\cdot)$ for the 2-D problem. For the 3-D problem, details are provided in [14]. The bilinear form $a(\cdot,\cdot)$ is given by

$$a(u,v) = \int_0^1 w_1(s)u'(s)v'(s)ds + \int_0^1 w_2(s)u''(s)v''(s)ds$$

and $L = L_v$ is

$$L_v(u) = \int_0^1 F(v(s,t))u(s)ds,$$

for the 2-D case.

2)Discrete Variational Problem: A well-known approach for solving such a problem is Galerkin's method, which consists in defining a similar discrete problem, over a finite-dimensional subspace V_h of the Sobolev space $H_0^2(\Omega)$. The associated discrete problem for (17) is:

1) Variational Problem 2: Find $v_h \in V_h$ such that

$$a(v_h, u_h) = L(u_h) \quad \forall \ u_h \in V_h. \tag{18}$$

A solution v_h of this discrete problem is an approximation of the solution v of the continuous variational problem.

This discrete problem leads to a finite linear system defined over the finite-dimensional space V_h.

The FEM provides an efficient tool for defining the space V_h.

3) The Finite-Element Method: The finite-element method is characterized by three aspects in the construction of the space V_h.

(FEM1) a tessellation is established over the parametrization set $\Omega = [0,1]$ ($[0,1] \times [0,1]$ in 3-D);

(FEM2) the functions $v_h \in V_h$ are typically piecewise polynomial; and

(FEM3) the basis of functions for the space V_h are chosen such that they have small support.

Hence, the FEM provides a finite dimensional space V_h and a discrete representation of the solution v_h approximating the solution v of the variational problem (17).

We use a conform finite-element method which insures that the space V_h is a subspace of $H_0^2(\Omega)$ and that the basis functions are $\mathcal{C}^1$ continuous. In the following we describe the choice of the subspace V_h in the 2-D and 3-D case.

4) The 2-D Curve Case: We consider a uniform subdivision of $\Omega = [0,1] = \bigcup_{i=1}^{N}[ih,(i+1)h]$, where N is the number of discretization points and $h = \frac{1}{N+1}$.

Since the variational problem 17 uses the space of admissible functions $H_0^2([0,1])$, the space V_h must satisfies $V_h \subset \mathcal{C}^1 \cap H_0^2([0,1])$ (for details see [8]). A choice for the subspace V_h is defined by:

$$V_h = \left\{ v \in \mathcal{C}^1([0,1]), v_{|[x_i,x_{i+1}]} \in P_3([x_i,x_{i+1}]) 0 \le i \le N \right\},$$

where $P_k(I)$ is the vector space of polynomials of degree k or less, restricted to the interval $I \subset \mathbb{R}$. We use the notation $v_{|I}$ to mean the restriction of the function v to the subset I. The space P_3 has been chosen since a polynomial $p \in P_3$ is uniquely determined by its values and the values of its first derivative at two distincts points. The basis functions of V_h are ϕ_i and Ψ_i, $1 \le i \le N$ defined by:

$$\phi_i(x_j) = \delta_{ij}, \quad \phi_i'(x_j) = 0 \quad 1 \le j \le N \tag{19}$$

$$\Psi_i(x_j) = 0, \quad \Psi_i'(x_j) = \delta_{ij} \quad 1 \le j \le N, \tag{20}$$

where:

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

Analytic expressions for the ϕ_i and Ψ_i are given in [14]; Fig. 2 shows plots of the functions.

A function $v_h \in V_h$ is completely defined by the values of v_h and v_h' at each of the nodes x_i, and we have the identity:

$$v_h = \sum_{i=1}^{N} v_h(x_i)\phi_i + \sum_{i=1}^{N} v_h'(x_i)\Psi_i, \tag{21}$$

yielding an expression for v_h in terms of a finite collection of unknowns.

Using the FEM and the above choices for the implementation in order to compute the solution to the 2-D balloon model, we obtain the results presented in Fig. 11.

5) The 3-D Surface Case: A tessellation of the domain Ω in 3-D and the construction of the subspace V_h using the Bogner–Fox–Schmit (BFS) elements are given in [14]. In this space, a function is completely determined by four values at each nodal point $a_{ij} = (ih_s, jh_r)$, specifically, the values for $v_h, \frac{\partial v_h}{\partial s}, \frac{\partial v_h}{\partial r}$ and $\frac{\partial^2 v_h}{\partial s \partial r}$. The corresponding basis functions are shown in Fig. 3. These basis functions can also be obtained through a tensorial product of the functions ϕ and Ψ (19–20).

The tesselation of the domain Ω could be done with triangular patches providing an adaptive mesh and requiring a bigger computational complexity (see [8] for a complete description of the different basis functions). The choice of the BFS elements is due to their ability to tesselate easily rectangular domains and mainly to the reduced number of neighboring nodes and degrees of freedom (four at each nodal point).

Expressing $v_h \in V_h$ in the BFS basis leads to the identity:

$$v_h = \sum_{i,j=0}^{N_s-1,N_r-1} v_h(a_{ij})\varphi_{ij} + \frac{\partial v_h}{\partial s}(a_{ij})\psi_{ij} + \frac{\partial v_h}{\partial r}(a_{ij})\eta_{ij} + \frac{\partial^2 v_h}{\partial s \partial r}(a_{ij})\zeta_{ij}, \tag{22}$$

which provides a C^1 function defined over the set Ω depending on a finite collection of parameters.

B. Discretization of the Problem

Once the space is discretized and the function v is represented as an element in a finite dimensional subspace, a linear system results:

$$A \cdot V = L, \tag{23}$$

where the matrix A is symmetric, positive definite and heptadiagonal (tridiagonal per bloc in the 3-D case) and V is the vector of coordinates of v_h in the chosen basis. These coordinates are in fact the values of v_h and its derivatives $\frac{\partial v_h}{\partial s}, \frac{\partial v_h}{\partial r}$, and $\frac{\partial^2 v_h}{\partial s \partial r}$ at the nodes of the tessellation.

Having discretized the problem (15) in space, we next have to discretize its variational formulation:
given $v_0 \in L^2(\Omega)$ and $F \in L^2(0, T, L^2(\Omega))$,
find a function $v \in L^2(0, T, H_0^2(\Omega)) \cap C^1(0, T, L^2(\Omega))$ satisfying:

$$\begin{cases} \frac{d}{dt}(v(t), \psi) + a(v(t), \psi) = L_v(\psi) \ \forall \psi \in H_0^2(\Omega) \\ v(0) = v_0 \\ w_1(s), w_2(s) \in L^\infty(\Omega), \quad w_1(s) \text{and } w_2(s) \geq \alpha > 0 \end{cases} \tag{24}$$

We then use finite differences in time. Finite differences in time may be viewed as as a way to formulate the following iterative method. We are only interested in the final result and so do not need an accurate solution in time. The result is simply:

$$\begin{cases} \frac{V^t - V^{t-1}}{\tau} + A \cdot V^t = L_{V^{t-1}} \\ V^0 = v_0 \text{ initial estimation,} \end{cases} \tag{25}$$

where τ is the time step. Equation (25) can be written in a form similar to the finite differences formulation (4), yielding

$$(\mathcal{I} + \tau A) \cdot V^t = V^{t-1} + \tau L_{V^{t-1}}, \tag{26}$$

which is the discrete version of (16). To solve the linear system $M \cdot V = N$ at each time step, for which the matrix $M = (\mathcal{I} + \tau A)$ is banded, symmetric and positive definite, we first note that M does not depend on t, and so its inverse may be precomputed using a Cholesky factorization.

Note that we assume here that the coefficient functions w_{ij} remain constant in time. If the coefficients do change in time, or even if they do not, an alternative method to solve this linear system is by means of a Conjugate Gradient method, in which the solution V^{t-1} is taken as an initial guess at time t. This approach appears to have a faster convergence than the Cholesky factorization method.

Remark that when using the FEM, the solution is less sensitive than with FDM to deformations of the mesh. It permits apparition of larger distances between neighboring nodes which happen, for example, when using the balloon model. However, like with the FDM, we periodically reparametrize the curve or surface, but without adding new node points. For a curve in 2-D, we construct a new parametrization using the existing curve by sampling at a regular distance between nodes, with a given number of nodes. For a surface in 3-D, we do the same as in 2-D in both directions of the parametrization, one direction after the other.

C. Performance and Complexity Analysis

The better complexity of the FEM was studied in 2-D and guided us to use it for the 3-D generalization of the model.

The FEM has a better complexity because, as compared to the FDM, the step size of the spatial discretization can be larger with the FEM, resulting in linear systems of smaller size. In general, we observe with the 2-D FDM model with our application that:

- if the step size is more than 2 pixels, then the curves passes over the edges or fails to be attracted to edges;
- for the balloon model, a dynamic reparametrization is often required since the length of the curve increases significantly during the time steps; and
- the size of the linear system is of the order of the length of the curve, due to the reparametrization.

While for the FDM, we follow the evolution of a set of points, with finite elements for the 2-D model, the curve which is between two points of the grid can deform, so that the image forces between two points are also considered. The computation of vector L (see Section IV-E) is made by numerical integration, interpolating along the interval $[(i-1)h, (i+1)h]$ for each node $s_i = ih$ of the subdivision. The numerical integration is made at the pixel size so that no information is lost.

If we compare results of FEM with those obtained using a finite-difference method (FDM) (as in [12]), we find out that, as expected, the finite-element method requires fewer points for the curve discretization and gives more stable results. This FEM gives also a faster convergence to an equilibrium.

Since the methods lead to the linear systems (25) for FEM or (4) for FDM, the algorithmic complexity can be deduced by examining the associated linear systems.

Let N be the number of the discretization points along the interval [0, 1]. For the FEM, the matrix A is a $2N \times 2N$ heptadiagonal array, while it is only $N \times N$ pentadiagonal for the FDM. In the FDM approach, the number of points N must be at least equal to the length l, in pixels, of the initial guess, and may have to increase in size. On the other hand, for the finite-element method, the number of points N is typically of the order of $l/6$. Thus, the matrix size for the FEM case is $\frac{2l}{6}$ by $\frac{2l}{6}$, which is 9 times smaller than with the FDM.

Moreover, with the FEM, the same number of nodes in the system is held fixed for all iterations. An initial computation of the inverse of the matrix A is sufficient for the whole process.

D. Elasticity and Rigidity Coefficients

The elasticity and rigidity coefficients w_{kl} play an important role in the convergence process of the surface toward the image edges. These coefficients must be chosen in a correct way such that the internal forces generated by terms of the energy E comprising the coefficients w_{kl} have the same magnitude as the external forces generated by the potential $P(v)$. Since a minimum of the energy E will involve a trade-off between the internal and external energy, the solution surface should fit the edge points while being smooth and regular. If the internal energy is preponderant, then the surface will tend to collapse on itself without detecting image edges, whereas if the external energy predominates, then the surface will converge along the image edge without any degree of smoothing.

To insure that both internal and external energy have the same order of magnitude, we have found it sufficient to choose the coefficients w_{kl} such that the linear system of (16) is well-conditioned. For example, the following assignments result in a well-conditioned system:

$$w_{10} = w_{01} = h_s^2 h_r^2 \text{ and } w_{20} = w_{11} = w_{02} = h_s^3 h_r^3,$$

where h_s and h_r are the discretization step of Ω. Setting $w_{10} = w_{01}$ and $w_{20} = w_{11} = w_{02}$ presupposes that the 3-D image data is isotropic, and thus that all directions have equal weight.

In [9], we propose a general approach for determining the regularizing parameters w_{ij} given an error margin on the accuracy of the reconstructed surface. This method allows also surface reconstruction preserving discontinuities.

E. The Computation of the Vector L

The computation of $L = (L(e_1), \cdots, L(e_N)))^T$ (where e_p is the chosen basis of V_h) depends upon $P = (x, y)$ which is known only at pixel (or voxel in 3-D) locations. The integrals $L(e_l) = \int_\Omega F\big(v^{t-1}(s,r)\big)e_l(s,r)dsdr$ represents the contribution of the external forces which cause the surface to be attracted toward the edges, and contribute to the linear system that we must solve at each iteration. Thus, the more we weight the potential P, the more closely the result tracks the edges and the faster is the convergence (at the expense of smoothness).

Since the potential P is known only at pixel locations, we must compute the $L(e_l)$ with a numerical integration. Consequently we compute ∇P at interpolated points $(x, y, z) \in \mathbb{R}^3$ by a trilinear interpolation of the eight neighbors.

To take into account all the contributions of the external forces, we modified the numerical integration formula such that every image point in the set $v([(i-1)h_s, (i+1)h_s] \times [(j-1)h_r, (j+1)h_r])$ is involved in the computation of each term $L(e_p)$. This method allows us to do an "adaptive subdivision" of the rectangle $K_{ij} = [ih_s, (i+1)h_s] \times [jh_r, (j+1)h_r]$ without adding nodal points and, consequently, without increasing the size of the linear system to be solved. This method significantly reduces the algorithmic complexity while increasing the accuracy and the convergence speed.

V. 3-D Results

Using a real 3-D deformable model to segment a 3-D image provides better results than the iterated application of a 2-D deformable model to successive 2-D cross-sections. In effect, the 3-D model easily bridges edge gaps in 3-D, i.e., not only within a cross section, but also between cross-sections, insuring that the result is globally a smooth surface.

Compared to the simplified approaches of Section III, the use of the full 3-D model to segment 3-D data significantly improves the robustness of the segmentation; for instance it is even possible to remove all the edges of a single cross section (assuming that the edges are correctly detected in other slices) without seriously degrading the final result. Fig. 12 shows the 3-D reconstruction obtained by using the 3-D balloon model of Section III applied to the data of Fig. 9. The final surface is more accurate and smoother than that obtained with the simplified approach shown in Fig. 10. On a Sun Sparc station, the result shown in Fig. 12 takes however about ten times more computation time than that required for the simplified approach.

Figs. 13 and 14 demonstrate another example of the 3-D model applied to artificial data. In this case, the initial surface is a cylinder (Fig. 13), where we have removed some edges in three successives cross sections, for comparison purposes. The deformable surface restores the missing edges and obtains a perfect reconstruction of the cylinder (Fig. 14), whereas a 2-D model fails due to the missing edges, even if we use the same attraction force as for the 3-D model.

Fig. 14 and also Figs. 17 and 18 show cross sections of the original 3-D image overlaid with the reconstructed surface on the same plane.

For our final example, we use real data: Image data of a part of a human head obtained with Magnetic Resonance Imaging (MRI). We make use of the "weight force" described in Section II-B-2, which allows us to begin with a much simpler initial surface than in [10]; In this case, the initial surface is a

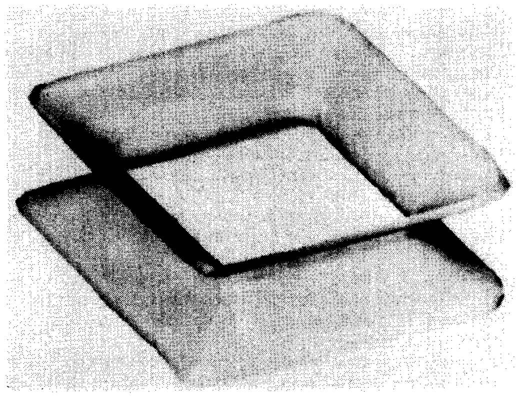


Fig. 12. Representation of the 3-D reconstructed surface using the data of Fig. 9.

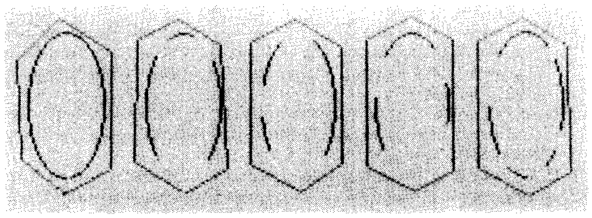


Fig. 13. Successive cross sections of the deteriorated edges and initial surface (in grey) given by the user.

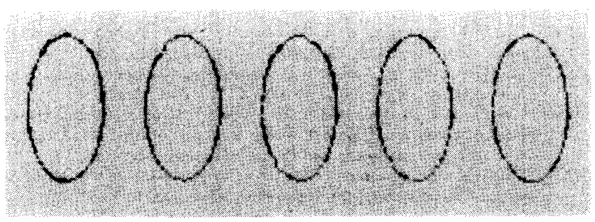


Fig. 14. Here, we show how the deformable surface (in grey) can reconstruct deteriorated edges by maintaining 3-D homogeneity. In this example, a 2-D model cannot reconstruct the missing edges even if we use a 3-D potential.

plane placed on one side of the 3-D image. The weight force makes the surface fall through the image until it is caught progressively by the shape of the face. The evolution of the surface is shown in Figs. 15 and 16. The final result is obtained after about 100 iterations. We show the final result overlaid on the original image data in vertical (Fig. 17) and horizontal (Fig. 18) cross sections. Here, by vertical and horizontal slices, we mean with respect to the representation of Fig. 16 We remark the accurate localization by the surface of the 3-D edges.

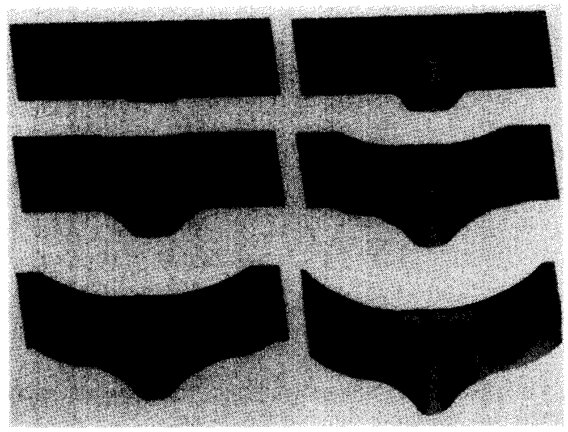


Fig. 15. Evolution of the 3-D surface "falling" on a 3-D MRI image of a head. The initial surface is a plane on the border of the image.

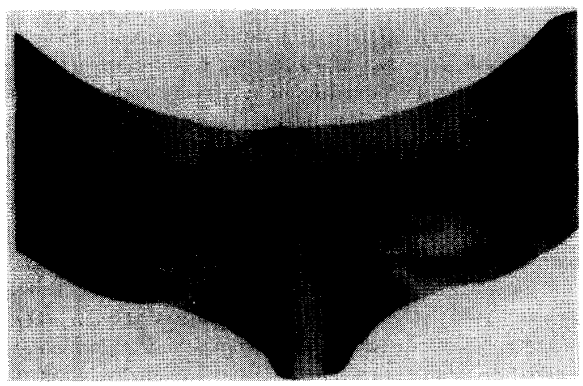


Fig. 16. Final result of the face.

Fig. 17. Overlays of some vertical cross sections of the final surface obtained by the algorithm with the original data.

VI. Conclusion

One of the goals of the regularization process in surface reconstruction is to obtain good estimates of partial derivatives of the surface in order to compute differential characteristics. Since the result of the deformable model reconstruction described here is an analytical description of class $\mathcal{C}^\infty$ almost everywhere (except along the borders of the finite elements, where the representation is only of class $\mathcal{C}^1$), we may compute, for example the first and second fundamental forms of the surface [18]. We could then extract a curvature-based primal sketch of the surface [6], [26], including intrinsic features such as parabolic lines, extrema of curvatures, and umbilic points, which can be used as landmarks for 3-D image interpretation [2]. We are conducting such a program [10], and will present results in a subsequent paper.

Another goal of this representation is the elastic matching of extracted features to an atlas, which is also the problem discussed in [3]. For this purpose, we would deform a curve or surface to best match the pattern using some measure of the distortion, such as the area between the two curves. This was

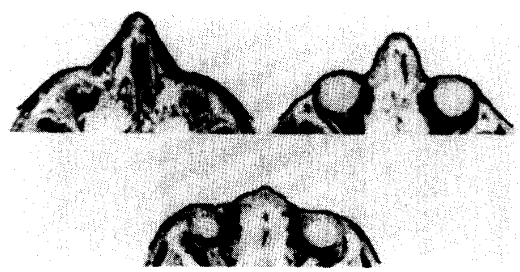


Fig. 18. Overlays of some horizontal cross sections of the final surface with the original data.

also studied in [35] with simple geometric shapes as templates which are deformed to match the image.

In order to achieve a representation on a deformable model, we have presented a 3-D generalization of the balloon model introduced in [12] which solves some of the problems encountered with the "snake" model of [23], [32]. We began with a survey of the use of an attraction potential generated by available edge data to reconstruct a curve or surface. We demonstrated some properties of deformable surfaces and their interaction with 3-D edge points. Our approach here extends an earlier approach, reported in [12], where we use a series of 2-D contours in successive cross sections to make a 3-D reconstruction of the surface of the ventricles. The simplified approach given here was implemented by defining a 3-D surface as a series of 2-D planar curves making a simultaneous and interdependent evolution, using the (9). The solution method used a finite-difference approach and an explicit scheme, which produced a fast computational algorithm.

We then implemented a finite-element method solution strategy, to solve the full 3-D problem. The reason for choosing a FEM approach as opposed to a finite-difference method was that:

1) the FEM approach requires fewer discretization points and consequently produces a smaller linear system to solve, thus reducing significantly the algorithmic complexity;
2) the FEM approach produces more accurate results, since the external forces can be computed more accurately;
3) the FEM approach provides an analytical representation of the surface.

To solve the full 3-D model of surface, we used a Bogner–Fox–Schmit finite rectangular element.

This method has been tested for several applications in medical image analysis. We showed promising results of our model on MR (magnetic resonance) images, to extract features like the contour of a face.

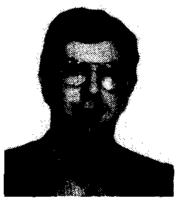
ACKNOWLEDGMENT

The authors would like to thank Nicholas Ayache and Robert Hummel for all their help with this paper.

REFERENCES

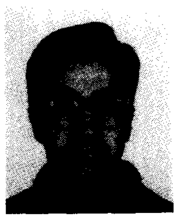
[1] N. Ayache, J. D. Boissonnat, E. Brunet, L. Cohen, J. P. Chièze, B. Geiger, O. Monga, J. M. Rocchisani, and P. Sander, "Building highly structured volume representations in 3-D medical images," in *Computer Aided Radiology*, Berlin, West Germany, June 1989.
[2] N. Ayache, J. D. Boissonnat, L. Cohen, B. Geiger, O. Monga, J. Levy-Vehel, and P. Sander, "Steps toward the automatic interpretation of 3-D images," *NATO ASI Series on 3-D Imaging in Medicine*, vol. F 60, pp. 107–120, 1990.
[3] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Comput., Vision, Graphics, and Image Processing*, vol. 46, pp. 1–21, 1989.
[4] A. Blake and A. Zisserman, *Visual Reconstruction.* Cmabridge, MA: The MIT Press, 1987.
[5] G. Borgefors, "Distance transformations in arbitrary dimensions," *Comput. Vision, Graphics, and Image Processing*, vol. 27, pp. 321–345, 1984.
[6] M. Brady, J. Ponce, A. Yuille, and H. Asada, "Describing surfaces," in *Proceedings of the Second International Symposium on Robotics Research*, H. Hanafusa and H. Inoue, Eds. Cambridge, MA: MIT Press, 1985, pp. 5–16.
[7] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[8] P. G. Ciarlet, *The finite element methods for elliptic problems.* Amsterdam, The Netherlands: North-Holland, 1987.
[9] I. Cohen, "Modèles déformables 2-D et 3-D: Application à la segmentation d' images médicales," Ph.D. thesis, Université Paris-IX Dauphine, June 1992.
[10] I. Cohen, L. D. Cohen, and N. Ayache, "Using deformable surface to segment 3-D images and infer differential structures, in *Proc. Second European Conf. Comput. Vision*, Santa Margherita Ligure, Italy, May 1992, pp. 648–652.
[11] L. D. Cohen, "On active contours models," in *Proceedings of the NATO ASI Active Perception and Robot Vision*, (Maratea, Italy, July 1989). New York: Springer.
[12] L. D. Cohen, "On active contour models and balloons," *Comput. Vision, Graphics, and Image Processing: Image Understanding*, vol. 53, no. 2, pp. 211–218, Mar. 1991.
[13] L. D. Cohen and I. Cohen, "A finite-element method applied to new active contour models and 3-D reconstruction from cross sections, in *Proc. Third Int. Conf. Computer Vision*, Osaka, Japan, Dec. 1990, pp. 587–591.
[14] ——, "finite-element methods for active contour models and balloons from 2-D to 3-D," Tech. Rep. 9124, Ceremade, Dec. 1991.
[15] P. E. Danielsson, "Euclidean distance mapping," *Comput. Vision, Graphics, and Image Processing*, vol. 14, pp. 227–248, 1980.
[16] H. Delingette, M. Hebert, and K. Ikeuchi, "Shape representation and image segmentation using deformable surfaces, in *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, Maui, HI, June 1991.
[17] R. Deriche, "Using canny's criteria to derive a recursively implemented optimal edge detector," *Int. J. Comput. Vision*, pp. 167–187, 1987.
[18] M. P. do Carmo, *Differential Geometry of Curves and Surfaces.* Englewood Cliffs, NJ: Prentice-Hall, 1976.
[19] P. Fua and Y. G. Leclerc, "Model driven edge detection," in *DARPA Image Understanding Workshop*, 1988.
[20] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems.* New York: Springer-Verlag, 1984.
[21] W. E. L. Grimson, *From Images to Surfaces: A computational study of the Human Early vision system.* Cambridge, MA: The MIT Press, 1981.
[22] I. L. Herlin and N. Ayache, "Features extraction and analysis methods for sequences of ultrasound images," in *Proceedings of the Second European Conference on Computer Vision 1992* Santa Margherita Ligure, Italy, May 1992.
[23] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision*, vol. 1, pp. 321–331, 1987.
[24] O. Monga and R. Deriche, "3-D edge detection using recursive filtering. Application to scanner images," in *IEEE Comput. Soc. Conf. Vision and Pattern Recognit.*, San Diego, CA, June 1989.
[25] T. Poggio, H. Voohrees, and A. Yuille, "A regularized solution to edge detection," Tech. Rep. A.I. Memo 833, MIT, May 1985.
[26] J. Ponce and M. Brady, "Toward a surface primal sketch," in *Proc., IJCAI*, 1985.
[27] D. Terzopoulos, "Image analysis using multigrid relaxation methods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 2, pp. 129–139, Mar. 1986.
[28] ——, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 4, pp. 413–424, July 1986.
[29] ——, "On matching deformable models to images," in *Topical Meeting on Machine Vision, Technical Digest Series.* New York: Optical Society of America, vol. 12, 1987, pp. 160–163.

[30] ——, "The computation of visible-surface representations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 4, pp. 417–438, July 1988.
[31] D. Terzopoulos, A. Witkin, and M. Kass, "Symmetry-seeking models for 3-D object reconstruction," *Int. J. Comput. Vision*, vol. 1, no. 3, pp. 211–221, Oct. 1987.
[32] ——, "Constraints on deformable models: recovering 3-D shape and nonrigid motion," *AI J.*, vol. 36, pp. 91–123, 1988.
[33] A. N. Tikhonov and V. Y. Arsenin, *Solutions of ill-posed problems.* New York: Winston, 1977.
[34] I. Weiss, "Shape reconstruction on a varying mesh," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 4, pp. 345–362, Apr. 1990.
[35] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates, in *Proc. Comput. Vision Pattern Recognit.*, San Diego, CA, June 1989.
[36] S. W. Zucker and R. M. Hummel, "A three-dimensional edge operator," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, no. 3, pp. 324–331, May 1981.



Laurent D. Cohen, was born in 1962. He was a student at the Ecole Normale Supérieure in Paris, France from 1981 to 1985. He received the Master's and Ph.D. degrees in applied mathematics from University of Paris 6, France, in 1983 and 1986, respectively. In 1990, he also obtained a second Ph.D. degree in computer science at Paris Sud University, Orsay, France.

From 1985 to 1987, he was member of the Computer Graphics and Image Processing group at Schlumberger Palo Alto Research, Palo Alto, CA, and Schlumberger Montrouge Research, Montrouge, France and remained consultant with Schlumberger afterwards. He began working with INRIA, Rocquencourt, France in 1988, and obtained in 1990 a position of Research Scholar with the French National Center for Scientific Research (CNRS) at CEREMADE, University Paris-Dauphine, Paris, France. His research interests and teaching at university are in applications of variational methods to image processing and computer vision, like deformable models, surface reconstruction, image segmentation, and restoration.



Isaac Cohen received the master of applied mathematics and automatics in 1989 and the Ph.D. degree in applied mathematics in 1992 from the University Paris-IX-Dauphine.

Since then, he has been a senior researcher at the EPIDAURE computer vision laboratory, Institut National de Recherche en Informatique et Automatique (INRIA), Rocquencourt, France. His current research interests are in the application of variational methods for surface reconstruction and nonrigid motion in computer vision and biomedical image analysis.