A Survey on Deformable Model and its Applications to Medical Imaging

Ravindra Hegadi
Dept. of Computer Science
Karnatak University
Dharwad, India

Arpana Kop
Dept. of Computer Science
Karnatak University
Dharwad, India

Mallikarjun Hangarge
Dept. of Computer Science
Karnataka College
Bidar, India

ABSTRACT
Deformable models provide a promising and vigorously researched model-based approach to computer-assisted medical image analysis. The widely recognized potency of deformable models stems from their ability to segment, match, and track images of anatomic structures by exploiting (bottom-up) constraints derived from the image data together with (top-down) a priori knowledge about the location, size, and shape of these structures. In this paper, a survey of deformable models and their latest extensions are presented.

General Terms
Image processing, medical image analysis.

Keywords
Deformable models, medical image, segmentation, active contours, level sets, GVF.

1. INTRODUCTION
The role of medical imaging has expanded beyond the simple visualization and inspection of anatomic structures. It has become a tool for surgical planning and simulation, intra-operative navigation, radiotherapy planning, and for tracking the progress of disease. For example, ascertaining the detailed shape and organization of anatomic structures enables a surgeon preoperatively to plan an optimal approach to some target structure. In radiotherapy, medical imaging allows the delivery of a necrotic dose of radiation to a tumor with minimal collateral damage to healthy tissue. Although modern imaging devices provide exceptional views of internal anatomy, the use of computers to quantify and analyze the embedded structures with accuracy and efficiency is limited. The shortcomings typical of sampled data, such as sampling artifacts, spatial aliasing, and noise, may cause the boundaries of structures to be indistinct and disconnected. The challenge is to extract boundary elements belonging to the same structure and integrate these elements into a coherent and consistent model of the structure. As a result, these model-free techniques usually require considerable amounts of expert intervention.
Deformable models, a promising and vigorously researched model-based approach to computer-assisted medical image analysis. The widely recognized potency of deformable models stems from their ability to segment, match, and track images of anatomic structures by exploiting (bottom-up) constraints derived from the image data together with (top-down) a priori knowledge about the location, size, and shape of these structures[1].

This work is supported by the grant under UGC Major Research Project FNO.34-104/2008(SR), Government of India.

1.1 Definition and its Functioning
Deformable models are curves or surfaces defined within an image domain that can move under the influence of internal forces, which are defined within the curve or surface itself, and external forces, which are computed from the image data. The internal forces are designed to keep the model smooth during the deformation. The external forces are defined to move the model toward an object boundary or other desired features within an image. By constraining extracted boundaries to be smooth and incorporating other priori information about the object shape, deformable models offer robustness to both image noise and boundary gaps and allow integrating boundary elements into a coherent and consistent mathematical description.

The idea of deforming a template for extracting image features dates back to the work of Fischler and Elschlager's spring-loaded templates [6] and Widrow's rubber mask technique [7].The popularity of Deformable Models is largely due to the influential paper "Snakes: Active Contours" by Kass, Witkin and Terzopoulous[3,61].The mathematical foundations of deformable models represent the confluence of geometry, physics, and approximation theory. Geometry serves to represent object shape, physics imposes constraints on how the shape may vary over space and time, and optimal approximation theory provides the formal underpinnings of mechanisms for fitting the models to measured data. The name "deformable models" stems primarily from the use of elasticity theory at the physical level, generally within a Lagrangian dynamics setting [4]. The physical interpretation views deformable models as elastic bodies which respond naturally to applied forces and constraints.

The energy grows monotonically as the model deforms away from a specified natural or "rest shape" and often includes terms that constrain the smoothness or symmetry of the model. In the Lagrangian setting, the deformation energy gives rise to elastic forces internal to the model. Taking a physics-based view of classical optimal approximation, external potential energy functions are defined in terms of the data of interest to which the model is to be fitted. These potential energies give rise to external forces which deform the model such that it fits the data.

1.2 Types of Representations
There are basically two types of deformable models: parametric deformable models (cf. [3, 8–10]) and geometric deformable models (cf. [11–14]). Parametric deformable models represent

curves and surfaces explicitly in their parametric forms during deformation. This representation allows direct interaction with the model and can lead to a compact representation for fast real-time implementation. Adaptation of the model topology, however, such as splitting or merging parts during the deformation, can be difficult using parametric models. Geometric deformable models, on the other hand, can handle topological changes naturally. These models, based on the theory of curve evolution [15–18] and the level set method [19, 20], represent curves and surfaces implicitly as a level set of a higher-dimensional scalar function. Their parameterizations are computed only after complete deformation, thereby allowing topological adaptivity to be easily accommodated. Despite this fundamental difference, the underlying principles of both methods are very similar. We first describe two different types of formulations for parametric deformable models: an energy minimizing formulation and a dynamic force formulation. Although these two formulations lead to similar results, the first formulation has the advantage that its solution satisfies a minimum principle whereas the second formulation has the flexibility of allowing the use of more general types of external forces. We then present several commonly used external forces that can effectively attract deformable models toward the desired image features.

2. PARAMETRIC DEFORMABLE MODEL

Different types of formulation are: (a) Energy minimizing formulation (b) Dynamic force formulation. Both lead to same results, but the first formulation has the advantage that its solution satisfies a minimum principle whereas the second formulation has the flexibility of allowing the use of more general types of external forces.

2.1 Energy minimizing Formulation

The basic premise of the energy minimizing formulation of deformable contours is to find a parameterized curve that minimizes the weighted sum of internal energy and potential energy. The internal energy specifies the tension or the smoothness of the contour. The potential energy is defined over the image domain and typically possesses local minima at the image intensity edges occurring at object boundaries. Minimizing the total energy yields internal forces and potential forces. Internal forces hold the curve together (elasticity forces) and keep it from bending too much (bending forces). External forces attract the curve toward the desired object boundaries. To find the object boundary, parametric curves are initialized within the image domain, and are forced to move toward the potential energy minima under the influence of both these forces.

To gain some insight about the physical behavior of deformable contours, we can view Eq. (4) as a force balance equation. Geometrically, a snake is a parametric contour embedded in the image plane $(x, y) = \Re^2$. The contour is represented as $v(s) = (x(s), y(s))^T$ where x and y are the coordinate functions and $s \in [0, 1]$ is the parametric domain. The shape of the contour subject to an image $I(x, y)$ is dictated by the functional

$$\varepsilon(v) = S(v) + P(v) \qquad (1)$$

The functional can be viewed as a representation of the energy of the contour and the final shape of the contour corresponds to the minimum of this energy. The first term of the functional,

$$s(v) = \int_0^1 w_1(s)\left|\frac{\partial v}{\partial x}\right|^2 + w_2(s)\left|\frac{\partial^2 v}{\partial x^2}\right|^2 ds \qquad (2)$$

is the internal deformation energy. It characterizes the deformation of a stretchy, flexible contour. Two physical parameter functions dictate the simulated physical characteristics of the contour: $w_1(s)$ controls the "tension" of the contour while $w_2(s)$ controls its "rigidity". The second term in (1) couples the snake to the image. Traditionally,

$$P(v) = \int_0^1 P(v(s))ds \qquad (3)$$

where $P(x, y)$ denotes a scalar potential function defined on the image plane. To apply snakes to images, external potentials are designed whose local minima coincide with intensity extrema, edges, and other features of interest [60].

The values of the non-negative functions $w_1(s)$ and $w_2(s)$ determine the extent to which the snake can stretch or bend at any point s on the snake. For example, increasing the magnitude of $w_1(s)$ increases the "tension" and tends to eliminate extraneous loops and ripples by reducing the length of the snake. Increasing $w_2(s)$ increases the bending "rigidity" of the snake and to make the snake smoother and less flexible. Setting the value of one or both of these functions to zero at a point s permits discontinuities in the contour at s.

$$\frac{\partial}{\partial s}\left(\alpha\frac{\partial X}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 X}{\partial s^2}\right) - \nabla P(X) = 0 \qquad (4)$$

which is

$$F_{int}(X) + F_{pot}(X) = 0 \qquad (5)$$

where internal force is

$$F_{\text{int}}(X) = \frac{\partial}{\partial s}\left(\alpha\frac{\partial X}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 X}{\partial s^2}\right) \qquad (6)$$

and the potential force is

$$F_{pot}(X) = -\nabla P(X) \qquad (7)$$

The internal force F_{int} discourages stretching and bending while the potential force F_{pot} pulls the contour toward the desired object boundaries. To find a solution to Eq. (4), the deformable contour is made dynamic by treating $X(s)$ as a function of time as well as — i.e $X(s, t)$. The partial derivative of X with respect to t is then set equal to the left-hand side of Eq. (4) as follows:

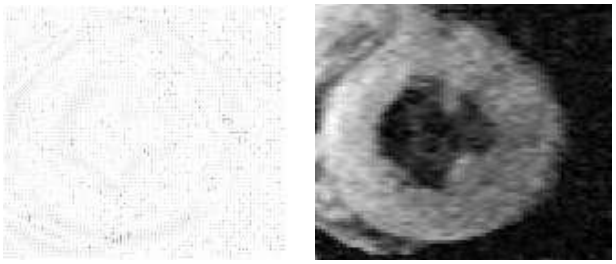
$$\gamma\frac{\partial X}{\partial t} = \frac{\partial}{\partial s}\left(\alpha\frac{\partial X}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 X}{\partial s^2}\right) - \nabla P(X) \qquad (8)$$

The coefficient γ is introduced to make the units on the left side consistent with the right side. When the solution $X(s, t)$

stabilizes, the left side vanishes and we achieve a solution of Eq. (4). We note that this approach of making the time derivative term vanish is equivalent to applying a gradient descent algorithm to find the local minimum of Eq. (1) [21]. Thus, the minimization is solved by placing an initial contour on the image domain and allowing it to deform according to Eq. (8). Figure 1 shows an example of recovering the left ventricle wall using Gaussian potential forces.

2.2 Dynamic force Formulation

In the previous section, the deformable model was modeled as a static problem, and an artificial variable t was introduced to minimize the energy. It is sometimes more convenient, however, to formulate the deformable model directly from a dynamic problem using a force formulation. Such a formulation permits the use of more general types of external forces that are not potential forces, i.e., forces that cannot be written as the negative



(a) (b)

Figure 1: An example of recovering the left ventricle wall using Gaussian potential forces. (a) Gaussian potential forces and (b) the result of applying Gaussian potential forces to a deformable contour, with a circular initial contour in gray and the final deformed contour in white.

gradient of potential energy functions. According to Newton's second law, the dynamics of a contour $X(s, t)$ must satisfy

$$\mu \frac{\partial^2 X}{\partial t^2} = F_{damp}(X) + F_{\text{int}}(X) + F_{ext}(X), \qquad (9)$$

where μ is a coefficient that has a mass unit and F_{damp} is the damping (or viscous) force defined as $-\gamma \partial X / \partial t$, with γ being the damping coefficient. In image segmentation, the mass coefficient μ in front of the inertial term is often set to zero since the inertial term may cause the contour to pass over the weak edges. The dynamics of the deformable contour without the inertial term becomes

$$\gamma \frac{\partial X}{\partial t} = F_{\text{int}}(X) + F_{ext}(X), \qquad (10)$$

The internal forces are the same as specified in Eq. (6). The external forces can be either potential forces or nonpotential forces. We note, however, nonpotential forces cannot be derived from the variational energy formulation of the previous section. An alternate variational principle does exist (see [22]); however, it is not physically intuitive.

External forces are often expressed as the superposition of several different forces:

$$F_{ext}(X) = F_1(X) + F_2(X) + \ldots + F_N(X), \qquad (11)$$

Where N is the total number of external forces. This superposition formulation allows the external forces to be broken down into more manageable terms. For example, one might define the external forces to be composed of both Gaussian potential forces and pressure forces, which are described in the next section.

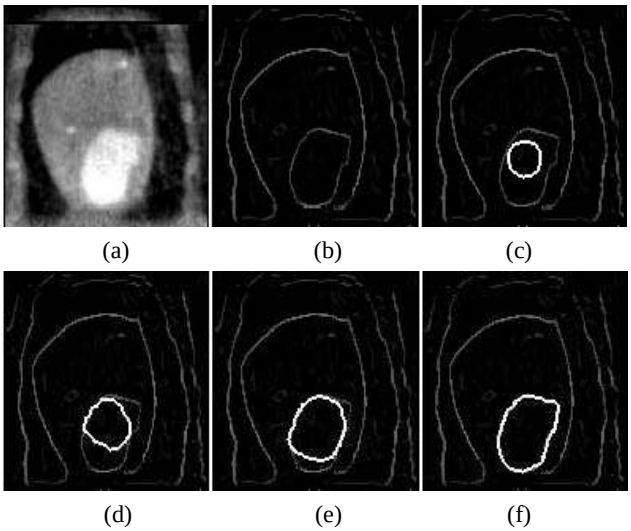
2.3 Types of External Forces

2.3.1 Multiscale Gaussian potential force

When this force was introduced and used, the force could only attract the model toward the boundary when it is initialized nearby. Terzopoulous, Witkin and Kass proposed using Gaussian potential forces at different scales to broaden its attraction range while maintaining the model's boundary localization accuracy. The basic idea is to first use a large value of σ is then reduced to create a potential energy function with a broad valley around the boundary. When the contour or snake reaches equilibrium, the value of σ is then reduced to allow tracking of the boundary at a finer scale. This scheme effectively extends the attraction range of the Gaussian potential force. Drawback of this approach is that there is no established theorem for how to schedule changes in σ.

2.3.2 Pressure force

The pressure force can either inflate or deflate the model. Hence it removes the requirement to initialize the model near the desired object boundary. Cohen suggested using another force together with Gaussian potential force. Models using pressure forces are also known as balloons. Defined as



(a) (b) (c)

(d) (e) (f)

Figure 2: An example of pressure forces driven deformable contours. (a) Intensity CT image slice of the left ventricle. (b) Edge detected image. (c) Initial deformable contour. (d) - (f) Deformable contour moving toward the left ventricle boundary, driven by inflating pressure force.

$$F_p(X) = w_p N(X), \quad (12)$$

Where $N(X)$ is the inward unit normal of the model at the point X and w_p is a constant weighting parameter. The sign of w_p determines whether to inflate or deflate the model and is typically chosen by the user. Recently, region information has been used to define with a spatial-varying sign based upon whether the model is inside or outside the desired object (see [23, 24]). The value w_p of determines the strength of the pressure force. It must be carefully selected so that the pressure force is slightly smaller than the Gaussian potential force at significant edges, but large enough to pass through weak or spurious edges. When the model deforms, the pressure force keeps inflating or deflating the model until it is stopped by the Gaussian potential force. An example of using deformable contour with an inflating pressure force is shown in Figure 2. A disadvantage in using pressure forces is that they may cause the deformable model to cross itself and form loops (cf. [25]).

2.3.3 Distance Potential Force

Another approach for extending attraction range is to define the potential energy function using a distance map as proposed by Cohen and Cohen [26]. The value of the distance map at each

(a) (b) (c)

Figure 3: An example of distance potential force field. (a) A U-shaped object, a close-up of its (b) boundary concavity, and (c) the distance potential force field within the concavity.

pixel is obtained by calculating the distance between the pixel and the closest boundary point, based either on Euclidean distance [27] or Chamfer distance [28]. By defining the potential energy function based on the distance map, one can obtain a potential force field that has a large attraction range. Given a computed distance map $d(x, y)$, one way of defining a corresponding potential energy, introduced in [26], is as follows:

$$P_d(x, y) = -w_d \exp[-d(x, y)^2] \quad (13)$$

The corresponding potential force field is given by $-\nabla P_d(x, y)$.

2.3.4 Gradient Vector Flow

The distance potential force is based on the principle that the model point should be attracted to the nearest edge points. This principle, however, can cause difficulties when deforming a contour or surface into boundary concavities [29]. A 2-D example is shown in Figure 3, where a U-shaped object and a close-up of its distance potential force field within the boundary concavity is depicted. Notice that at the concavity, distance potential forces point horizontally in opposite directions, thus preventing the contour from converging into the boundary concavity. To address this problem, Xu and Prince [2, 29] employed a vector diffusion equation that diffuses the gradient of an edge map in regions distant from the boundary, yielding a different force field called the gradient vector flow (GVF) field. The amount of diffusion adapts according to the strength of edges to avoid distorting object boundaries. A GVF field is defined as the equilibrium solution to the following vector partial differential equation:

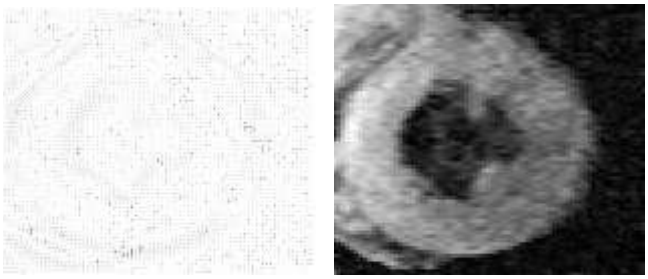
$$\frac{\partial v}{\partial t} = g(|\nabla f|)\nabla^2 v - h(|\nabla f|)(v - \nabla f), \quad (14)$$

Where $v(x, y, 0) = \nabla f, \partial v/\partial t$ denotes the partial derivative of $v(x, y, t)$ with respect to t, ∇^2 is the Laplacian operator (applied to each spatial component of v separately), and f is an edge map that has higher value at the desired object boundary and can be derived using any edge detector. The definition of the GVF field is valid for any dimension. Two examples of $g(r)$ and $h(r)$ are

$$g(r) = \exp -(r / k)^2$$

$$h(r) = 1 - g(r),$$

Where k is a scalar and r is a dummy variable, or $g(r) = \mu$ and $h(r) = r^2$, where μ is a positive scalar. GVF has been shown to have a large attraction range and improved convergence for deforming contours into boundary concavities [15, 29]. An



(a) **(b)**

Figure 4: An example of gradient vector flow driven deformable model contours. (a) A gradient vector flow field and (b) the result of applying gradient vector flow force to a deformable contour, with the circular initial contour shown in gray and the final deformed contour in white.

example of using a GVF force field is shown in Figure 4.

2.3.5 Dynamic Distance Force

An external force that is similar to distance potential force but does not possess the boundary concavity problem has been proposed [44, 45]. This approach derives an external force by computing a signed distance at each point on the deformable contour or surface. This signed distance is calculated by determining the closest boundary point or other image feature along the model's normal direction. The distance values are recomputed each time the model is deformed. Several criteria can be used to define the desired boundary point to be searched. The most common one is to use image pixels that have a high image intensity gradient magnitude or edge points generated by an edge detector. A threshold is specified for the maximum search distance to avoid confusion with outliers and to reduce the

computation time. The resulting force, which is called as the dynamic distance force, can attract deformable models to the desired image feature from a fairly long range limited only by the threshold.

Given a point X on the contour or surface, its inward unit normal $N(X)$, the computed signed distance $D(X)$, and a specified distance threshold D_{max}, a typical definition for the dynamic distance force is

$$F_D(X) = w_D \frac{D(X)}{D_{\max}} N(X) \tag{15}$$

The weakness of this method is that a relatively time-consuming one dimensional search along the normal direction must be performed each time the model deforms. Setting the search distance threshold lower can reduce the run time but has the undesirable side effect of decreasing the attraction range of the dynamic distance force.

2.3.6 Interactive Force

In many clinical situations, it is important to allow an operator to interact with the deformable model as it is deforming. This interaction improves the accuracy of the segmentation result when automated external forces fail to deform the model to the desired feature in certain regions. For example, the user may want to pull the model toward significant image features, or would like to constrain the model so that it must pass through a set of landmark points identified by an expert. De-formable models allow these kinds of user interactions to be conveniently modelled as additional force terms.

Two kinds of commonly used interactive forces are spring forces and volcano forces, proposed by Kass et al. [3]. Spring forces are defined to be proportional to the distance between a point x on the model and a user-specified point p:

$$F_s = w_s(p - X). \tag{16}$$

Spring forces act to pull the model toward p. The further away the model is from p, the stronger the pulling force. The point x is selected by finding the closest point on the model to p using a heuristic search around a local neighbourhood of p. Volcano forces are designed to push the model away from a local region around a "volcano" point p. For computational efficiency, the force is only computed in a neighbourhood $N(p)$ as follows:

$$F_v(X) = \begin{cases} w_v \dfrac{r}{|r|^3} & X \in N(p) \\ 0 & X \notin N(p) \end{cases} \tag{17}$$

Where $r = X - p$. Note that the magnitude of the forces is limited near $r = 0$ to avoid numerical instability. Another possible definition for volcano forces is

$$F_v(X) = \begin{cases} w_v \exp\left(-\dfrac{|r|^2}{\sigma_v^2}\right)\dfrac{r}{|r|} & X \in N(p) \\ 0 & X \notin N(p) \end{cases}$$

where σ_v is used to adjust the strength distribution of the volcano force.

2.4 Numerical Implementation:

Various numerical implementations of deformable models have been reported in the literature. For examples, the finite difference method [3], dynamic programming [8], and greedy algorithm [30] have been used to implement deformable contours, while finite difference methods [5] and finite element methods [10, 21, 31] have been used to implement deformable surfaces. The finite difference method requires only local operations and is efficient to compute. The finite element method, on the other hand, is more costly to compute but has the advantage of being well adapted to the irregular mesh representations of deformable surfaces.

So far, we have formulated the deformable model as a continuous curve or surface. In practice, however, it is sometimes more straightforward to design the deformable models from a discrete point of view. Example of work in this area includes [32–37].

Parametric deformable models have been applied successfully in a wide range of applications; however, they have two main limitations. First, in situations where the initial model and the desired object boundary differ greatly in size and shape, the model must be reparameterized dynamically to faithfully recover the object boundary. Methods for reparameterization in 2D are usually straightforward and require moderate computational overhead. Reparameterization in 3D, however, requires complicated and computationally expensive methods. The second limitation with the parametric approach is that it has difficulty dealing with topological adaptation such as splitting or merging model parts, a useful property for recovering either multiple objects or an object with unknown topology. This difficulty is caused by the fact that a new parameterization must be constructed whenever the topology change occurs, which requires sophisticated schemes [38, 39].

3. GEOMETRIC DEFORMABLE MODEL

Geometric deformable models, proposed independently by Caselles et al. [11] and Malladi et al. [12], provide an elegant solution to address the primary limitations of parametric deformable models. These models are based on curve evolution theory [15–18] and the level set method [19, 20]. In particular, curves and surfaces are evolved using only geometric measures, resulting in an evolution that is independent of the parameterization. As in parametric deformable models, the evolution is coupled with the image data to recover object boundaries. Since the evolution is independent of the parameterization, the evolving curves and surfaces can be represented implicitly as a level set of a higher-dimensional function. As a result, topology changes can be handled automatically.

In this section, we first review the fundamental concepts in curve evolution theory and the level set method. We next present three types of geometric deformable models, and the difference being in the design of speed functions. We then show a mathematical relationship between a particular class of parametric and geometric models. Next, we describe a numerical

implementation of geometric deformable models proposed by Osher and Sethian [19]. Finally, at the end of this section we compare geometric deformable models with parametric deformable models. We note that although the geometric deformable models are presented in 2D, their formulation can be directly extended to 3D. A thorough treatment on evolving curves and surfaces using the level set representation can be found in [20].

3.1 Curve Evolution Theory

The purpose of curve evolution theory is to study the deformation of curves using only geometric measures such as the unit normal and curvature as opposed to the quantities that depend on parameters such as the derivatives of an arbitrary parameterized curve. Let us consider a moving curve, $X(s, t) = [X(s, t), Y(s, t)]$, where s is any parameterization and t is the time, and denote its inward unit normal as N and its curvature as κ, respectively. The evolution of the curve along its normal direction can be characterized by the following partial differential equation:

$$\frac{\partial X}{\partial t} = V(\kappa)N, \tag{18}$$

where $V(\kappa)$ is called the speed function, since it determines the speed of the curve evolution. We note that a curve moving in some arbitrary direction can always be reparameterized to have the same form as Eq. (18) [40]. The intuition behind this fact is that the tangent deformation affects only the curve's parameterization, not its shape and geometry. The most extensively studied curve deformations in curve evolution theory are *curvature deformation* and *constant deformation*. Curvature deformation is given by the so-called *geometric heat equation*: $\frac{\partial X}{\partial t} = \alpha\kappa N$, where α is a positive constant. This equation will smooth a curve, eventually shrinking it to a circular point [41]. The use of the curvature deformation has an effect similar to the use of the elastic internal force in parametric deformable models.

Constant deformation is given by $\frac{\partial X}{\partial t} = V_0 N$, where V_0 is a coefficient determining the speed and direction of deformation. Constant deformation plays the same role as the pressure force in parametric deformable models. The properties of curvature deformation and constant deformation are complementary to each other. Curvature deformation removes singularities by smoothing the curve, while constant deformation can create singularities from an initially smooth curve. The basic idea of the geometric deformable model is to couple the speed of deformation (using curvature and/or constant deformation) with the image data, so that the evolution of the curve stops at object boundaries. The evolution is implemented using the level set method. Thus, most of the research in geometric deformable models has been focused in the design of speed functions.

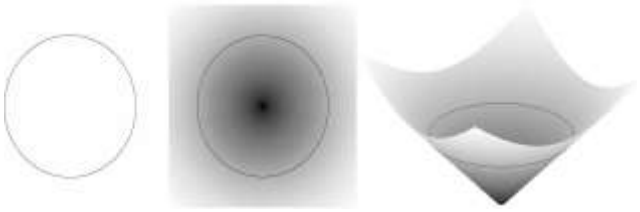


Figure 5: An example of embedding a curve as a level set. (a) A single curve. (b) The level set function where the curve is embedded as the zero level set (in black). (c) The height map of the level set function with its zero level set depicted in black.

3.2 Level Set Method

We now review the level set method for implementing curve evolution. The level set method is used to account for automatic topology adaptation, and it also provides the basis for a numerical scheme that is used by geometric deformable models. The level set method for evolving curves is due to Osher and Sethian [19, 42, 43]. In the level set method, the curve is represented implicitly as a level set of a 2D scalar function — referred to as the level set function — which is usually defined on the same domain as the image. The level set is defined as the set of points that have the same function value. Figure 5 shows an example of embedding a curve as a zero level set. It is worth noting that the level set function is different from the level sets of images, which are sometimes used for image enhancement [44]. The sole purpose of the level set function is to provide an implicit representation of the evolving curve.

Instead of tracking a curve through time, the level set method evolves a curve by updating the level set function at fixed coordinates through time. This perspective is similar to that of a Euclidean formulation of motion as opposed to a Lagrangian formulation, which is analogous to the parametric deformable model. A useful property of this approach is that the level set function remains a valid function while the embedded curve can change its topology. This situation is depicted in Figure 6.

We now derive the level set embedding of the curve evolution Eq. (18). Given a level set function $\phi(x, y, t)$ with the contour $X(s, t)$ as its zero level set, we have

$$\phi[X(s. t), t] = 0.$$

Differentiating the above equation with respect to t and using the chain rule, we obtain

$$\frac{\partial \phi}{\partial t} + \nabla\phi . \frac{\partial X}{\partial t} = 0. \tag{19}$$

where $\nabla\phi$ denotes the gradient of ϕ. We assume that ϕ is negative inside the zero level set and positive outside. Accordingly, the inward unit normal to the level set curve is given by $N = -\frac{\nabla\phi}{|\nabla\phi|}$.

Three issues need to be considered in order to implement geometric deformable contours:

1. An initial function $\phi(x, y, t) = 0$ must be constructed such that its zero level set corresponds to the position of the initial contour. A common choice is to set $\phi(x, y, 0) = D(x, y)$, where $D(x, y)$, is the signed distance from each grid point to the zero level set. The computation of the signed distance for an arbitrary initial curve is expensive. Recently, Sethian and Malladi developed a method called the fast marching

method, which can construct the signed distance function in O (N *log* N), N where is the number of pixels. Certain situations may arise, however, where the distance may be computed much more efficiently. For example, when the zero level set can be described by the exterior boundary of the union of a collection of disks, the signed distance function can be computed in O (N) as

$$D(x) = \min_{i=1,2,...,M} (|x - c_i| - r_i),$$

Where $\boldsymbol{x} = (x, y)$, M is the number of initial disks, $\boldsymbol{c}_i$ and r_i are the center and radius of each disk.

2. Since the evolution equation is derived for the zero level set only, the speed function V (κ), in general, is not defined on other level sets. Hence, we need a method to extend the speed function V (κ) to all of the level sets. We note that the expressions for the unit normal and the curvature, however, hold for all level sets. Many approaches for such extensions have been developed (see [20] for a detailed discussion on this topic). However, the level set function that evolves using these extended speed functions can lose its property of being a signed distance function, causing inaccuracy in curvature and normal calculations. As a result, reinitialization of the level set function to a signed distance function is often required for these schemes. Recently, a method that does not suffer from this problem was proposed by Adalsteinsson and Sethian [45]. This method casts the speed extension problem as a boundary value problem, which can then be solved efficiently using the fast marching method.

3. In the application of geometric contours, constant deformation is often used to account for large-scale deformation and narrow boundary indentation and protrusion recovery. Constant deformation, however, can cause the formation of sharp corners from an initial smooth zero level set. Once the corner is developed, it is not clear how to continue the deformation, since the definition of the normal direction becomes ambiguous. A natural way to continue the deformation is to impose the so-called entropy condition originally proposed in the area of interface propagation by Sethian [46].

4. VARIATIONS OR EXTENSIONS TO DEFORMABLE MODELS

Numerous extensions have been proposed to the deformable models described in the previous sections, particularly to extend the parametric deformable models. These extensions address two major areas for improving standard deformable models. The first area is the incorporation of additional prior knowledge into the models. Use of prior knowledge in a deformable model can lead to more robust and accurate results. This is especially true in applications where a particular structure that requires delineation has similar shape across a large number of subjects. Incorporation of prior knowledge requires a training step that involves manual interaction to accumulate information on the variability of the object shape being delineated. This information is then used to constrain the actual deformation of the contour or surface to extract shapes consistent with the training data.

The second area that has been addressed by various extensions of deformable models is in modeling global shape properties. Traditional parametric and geometric deformable models are local models — contours or surfaces are assumed to be locally smooth. Global properties such as orientation and size are not explicitly modeled. Modeling of global properties can provide greater robustness to initialization. Furthermore, global properties are important in object recognition and image interpretation applications because they can be characterized using only a few parameters. Note that although prior knowledge and global shape properties are distinct concepts, they are often used in conjunction with one another. Global properties tend to be much more stable than local properties. Therefore, if information about the global properties is known a priori, it can be used to greatly improve the performance of the deformable model. In this section, we review several extensions of deformable models that use prior knowledge and/or global shape properties. We focus on revealing the fundamental principles of each extension and refer the reader to the cited literature for a full treatment of the topic.

4.1 Deformable Fourier models

In standard deformable models, a direct parameterization is typically utilized for representing curves and surfaces. Staib and Duncan [47] have proposed using a Fourier representation for parameterizing deformable contours and surfaces. A Fourier representation for a closed contour is expressed as

$$X(s) = \begin{bmatrix} X(s) \\ Y(s) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos 2\pi ks \\ \sin 2\pi ks \end{bmatrix}, \qquad (20)$$

where a_0, c_0, a_1, b_1, c_1, d_1, … are Fourier coefficients. The Fourier coefficients of X (s) are computed by

$$a_0 = \frac{1}{2\pi}\int_0^1 X(s)ds, \qquad a_k = \frac{1}{\pi}\int_0^1 X(s)\cos 2\pi ks\, ds,$$

$$b_k = \frac{1}{\pi}\int_0^1 X(s)\sin 2\pi ks\, ds$$

Coefficients of Y(s) are computed in analogous fashion. The advantages of the Fourier representation are that a compact representation of smooth shapes can be obtained by truncating the series and that a geometric description of the shape can be derived to characterize global shape properties. From Eq. (20), the coefficients a_0 and c_0 define the translation of the contour. Each subsequent term in the series expansion follows the parametric form of an ellipse. It is possible to map the coefficients to a parameter set that describes the object shape in terms of standard properties of ellipses [47]. Furthermore, like the Fourier coefficients, these parameters follow a scale ordering, where low index parameters describe global properties and higher indexed parameters describe more local deformations.

Staib and Duncan apply a Bayesian approach to incorporating prior information into their model. A prior probability function is defined by first manually or semi-automatically delineating structures of the same class as the structure to be extracted. Next, these structures are parameterized using the Fourier coefficients, or using the converted parameter set based on ellipses. Mean and

variance statistics are finally computed for each of the parameters.

Assuming independence between the parameters, the multivariate Gaussian prior probability function is given by

$$Pr(p) = \prod \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(p_i-\mu_i)^2}{2\sigma_i^2}} \quad (21)$$

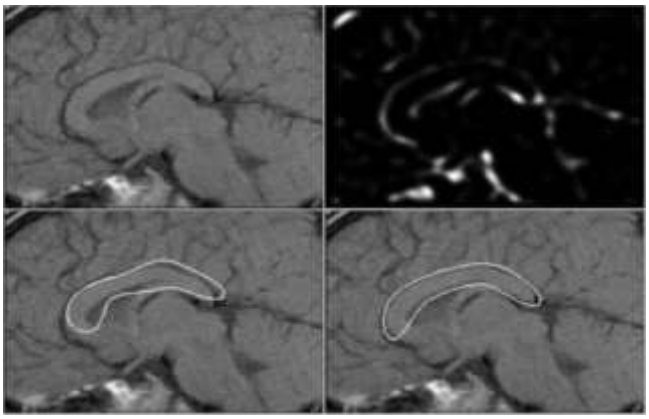


Figure 7: Segmenting the corpus callosum from an MR midbrain sagittal image using a deformable Fourier model. Top left: MR image (146X106). Top right: positive magnitude of the Laplacian of the Gaussian (σ = 2.2). Bottom left: initial contour (six harmonics). Bottom right: final contour on the corpus callosum of the brain.

where $p = (p_1, p_2, ..., p_N)$ is the parameter vector derived by truncating the Fourier coefficients, μ_i is the mean of the *i*th parameter in the training data, and σ_i^2 is the variance. A posterior probability function is defined that balances the prior probability model and a data model, which measures the discrepancy between boundary features in the image and the deformable contour. In [47], a gradient ascent method was used to maximize the posterior probability function. More recently, a genetic algorithm was proposed in [48]. Figure 7 shows an example of using the deformable Fourier model to recover the corpus callosum of the human brain.

4.2 Deformable models using modal analysis

Another way to restrict the mostly unstructured motion associated with the standard deformable model is to use modal analysis (Pentland and Horowitz [49], Nastar and Ayache [53]). This approach is similar to the deformable Fourier model except that both the basis functions and the nominal values of their coefficients are derived from a template object shape.

Deformable models based on modal analysis use the theory of finite elements [50]. An object is assumed to be represented by a finite set of *elements* whose positions are defined by the positions of *n* nodes, which are points in *d* -dimensional space. The node positions can be stacked into a vector ***X***, which has length *nd*, and element interpolation characterizes the complete object shape on the continuum. If the object moves or deforms, its new position is given by ***X***+***U***, where ***U*** is a vector of length *nd* representing the collection of nodal displacements.

The equation governing the object's motion can be written as a collection of ordinary differential equations constraining the nodal displacements. This is compactly written as

$$M\frac{d^2U}{dt^2} + C\frac{dU}{dt} + KU = f,$$

where ***M***, ***C***, and ***K*** are the mass, damping and stiffness matrices of the system and ***f*** is a *nd*-dimensional vector of external forces acting on the nodes. Both ***U*** and ***f*** are assumed to be functions of time.

4.3 Deformable superquadrics

Another extension of deformable models that has been used for incorporating local and global shape features is the *deformable superquadric*, proposed by Terzopoulos and Metaxas [31]. This is essentially a hybrid technique where a superquadric surface, which can be defined with a relatively small number of parameters, is allowed to deform locally for reconstructing the shape of an object. Although the fitting of global and local deformations is performed simultaneously, the global deformation is forced to account for as much of the object shape as possible. The estimated superquadric therefore captures the global shape characteristics and can readily be used in object recognition applications, while the local deformations capture the details of the object shape.

Terzopoulos and Metaxas consider models that are closed surfaces, denoted by ***x*** (***u***), where the parametric coordinates

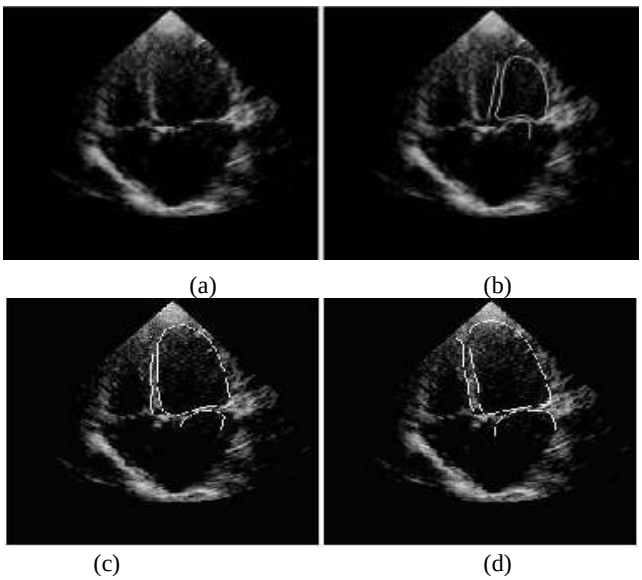


Figure 9: An example of Active Shape Models. (a) An echocardiogram image. (b) The initial position of the heart chamber boundary model. The location of the model after (c) 80 and (d) 200 iterations.

$\boldsymbol{u}$=[u, v]. This surface can be expressed as

$$\boldsymbol{x}(\boldsymbol{u}) = \boldsymbol{c} + \boldsymbol{R}\boldsymbol{p}(\boldsymbol{u}), \quad (22)$$

where $\boldsymbol{c}$ is a translation vector, and $\boldsymbol{R}$ is a rotation matrix. The vector function $\boldsymbol{p}(\boldsymbol{u})$ denotes the model shape irrespective of pose and can further be expressed as

$$\boldsymbol{p}(\boldsymbol{u}) = \boldsymbol{s}(\boldsymbol{u}) + \boldsymbol{d}(\boldsymbol{u}) \quad (23)$$

where $\boldsymbol{s}(\boldsymbol{u})$ is a reference shape consisting of the low parameter global shape model, and $\boldsymbol{d}(\boldsymbol{u})$ is a displacement function consisting of the local deformations. The reference shapes in this case are superquadrics, which are an extension of standard quadric surfaces. These surfaces have been used in a variety of applications for computer graphics and computer vision, because of their ability to accommodate a large number of shapes with relatively few parameters. The kind of superquadric of interest here is the superellipsoid, which can be expressed implicitly as in [51].

4.4 Active shape models (ASMs)

Active shape models (ASMs) proposed by Cootes et al. [52, 53] use a different approach to incorporate prior shape information. Their prior models are not based on the parameterization, but are instead based on a set of points defined at various features in the image. In the following, we summarize how the prior model is constructed and used to enhance the performance of a deformable model and how the ASM paradigm can be extended to incorporate prior information on the image intensity rather than on the shape alone.

4.4.1 Construction of the ASM prior model

The ASM prior model is constructed by first establishing a set of labeled point features, or landmarks, within the class of images to be processed [see Figs. 8(a) and (b)]. These points are manually selected on each of the images in the training set. Once selected, the set of points for each image is aligned to one another with respect to translation, rotation, and scaling. This is accomplished using an iterative algorithm based on the Procrustes method [54]. This linear alignment allows studying the object shape in a common coordinate frame, which we will refer to as the model space of the ASM. After the alignment, there is typically still a substantial amount of variability in the coordinates of each point. To compactly describe this variability as a prior model, Cootes and Taylor developed the Point Distribution Model (PDM), which we now describe. Given N aligned shapes $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, $\boldsymbol{Y}_N$ in the model space, where $\boldsymbol{Y}_i = (x_{i0}, y_{i0} \ldots x_{in-1}, y_{in-1})^T$ is a $2n$-dimensional vector describing the coordinates of the n points from the ith shape, the mean shape, $\bar{\boldsymbol{Y}}$, is defined to be

$$\bar{\boldsymbol{Y}} = \frac{1}{N}\sum_{i=1}^{N} Y_i . \quad (24)$$

The covariance matrix, S, is computed by

$$S = \frac{1}{N-1}\sum_{i=1}^{N} (Y_i - Y)(Y_i - Y)^T$$

The eigenvectors corresponding to the largest eigenvalues of the covariance matrix describe the most significant modes of variation. Because almost all of the variability in the model can be described using these eigenvectors, only m such eigenvectors are selected to characterize the entire variability of the training set. Note that in general m is significantly smaller than the number of points in the model.

4.4.2 Model fitting procedure

The key idea of ASMs is to constrain the behavior of deformable models using the PDM obtained as described in the previous section (cf. [53, 55, 56]). At each iteration, a standard deformation of the parametric deformable model is approximated by adjusting both the pose (translation, rotation, and scale) parameters and the shape parameters of the model instance. Thus, only deformations that produce shapes similar to those in the training set are allowed. The iteration stops when changes in both the pose and shape parameters are insignificant. Figure 9 shows an example of using active shape models to extract the heart wall from an ultrasound image.

Let us denote the position of the model instance at the beginning of a deformation step as $\boldsymbol{X} = (X_0, Y_0, \ldots, X_{n-1}, Y_{n-1})^T$ and the required deformation computed from both internal and external forces as a displacement vector $d\boldsymbol{X} = (dX_0, dY_0, \ldots, dX_{n-1}, dY_{n-1})^T$. Then the position of the model instance, $\boldsymbol{X}$, can be compactly represented by its pose and shape parameters, i.e.,

$$\boldsymbol{Y} = \bar{\boldsymbol{Y}} + \boldsymbol{P}\boldsymbol{b}, \text{ and}$$

$$\boldsymbol{X} = M(s, \theta)[\boldsymbol{Y}] + \boldsymbol{X}_c \quad (25)$$

where s is the scaling factor, θ is rotational angle, $M(s, \theta)[\boldsymbol{Y}]$ is a linear transformation that performs scaling and rotation on $\boldsymbol{Y}$, and $\boldsymbol{X}_c = (X_c, Y_c)$ is the center of the model instance. First, a global fit is performed by adjusting the pose parameters so that the generated model instance aligns best with the expected model instance $\boldsymbol{X} + d\boldsymbol{X}$. The proper pose parameter adjustments, ds, $d\theta$, and $d\boldsymbol{X}_c$, can be estimated efficiently using a standard least-squares approach (see [53] for details).

4.5 Other models

Additional extensions have also been proposed to use global shape information or prior shape information. For example, Ip and Shen [57] incorporated prior shape information by using an affine transformation to align a shape template with the deformable model and guide the model's deformation to produce a shape consistent with the template.

The deformable Fourier model, active shape model, and other extensions we discussed so far are all parametric deformable models. Guo and Vemuri[58] have proposed a framework for incorporating global shape prior information into geometric deformable models. Like the deformable superquadric, their hybrid geometric deformable model uses a combination of an underlying, low parameter, generator shape that is allowed to evolve. Their model thus retains the advantages of traditional geometric deformable models, such as topological adaptivity. External forces for deformable models are typically defined from edges in the image. Fritsch et al. [59] have developed a technique called deformable shape loci, which uses information on the medial loci or cores of the shapes to be extracted. The incorporation of cores provides greater robustness to image disturbances such as noise and blurring than purely edge-based models. This allows their model to be fairly robust to initialization as well as imaging artefacts. They also employed a probabilistic prior model for important shape features as well as for the spatial relationships between these features.

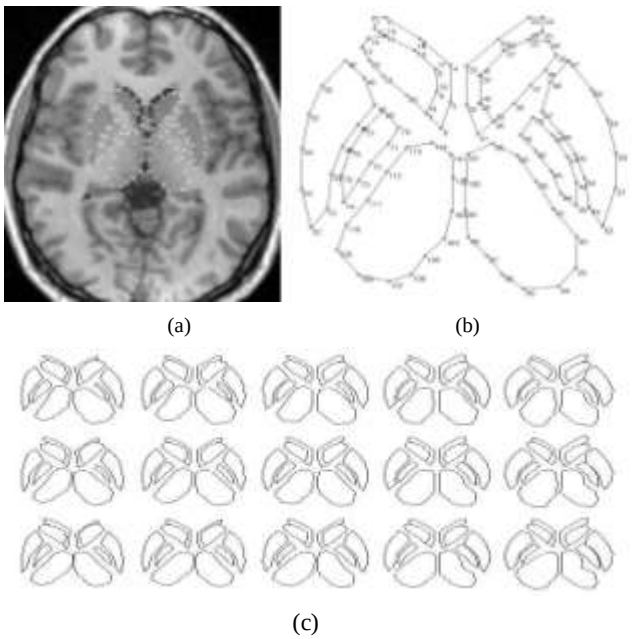


Figure 8: An example of constructing Point Distribution Models. (a) An MR brain image, transaxial slice, with 114 landmark points of deep neuroanatomical structures superimposed. (b) A 114-point shape model of 10 brain structures. (c) Effect of simultaneously varying the model's parameters corresponding to the first two largest eigenvalues (on a bi-dimensional grid).

5. CONCLUSION

We have described the fundamental formulation of both parametric and geometric deformable models and shown that they can be used in recovering shape boundaries. We have also derived an explicit mathematical relationship between these two formulations that allows one to share the design of external forces and speed functions. This may lead to new, improved deformable models. Finally, we give a brief overview of several important extensions of de-formable models that use application-specific prior knowledge and/or global shape properties to obtain more robust and accurate results.

We expect that further improvements in deformable models will be made by the continued research in external force and speed function design, model representation, model training and learning, and model performance validation. Another challenging research direction is to develop deformable models that have a greater control in topology. For example, models that can both constrain or change topology depending on the requirements of an application would be extremely useful. Promising approaches have been proposed recently, such as the work by McInerney and Terzopoulos [39], who developed a hybrid method that maintains both implicit and explicit representation for a given model to allow more effective control of the topology. Finally, integrating deformable models with existing medical systems, such as surgical simulation, planning, and treatment systems, can further validate the application of deformable models in a clinical setting and may in turn stimulate the development of better deformable models.

6. REFERENCES

[1] Tim McInerney and Demetri Terzopoulous, Dept of Computer Science, University of Toronto, "Deformable Models in Medical Image Analysis: A survey", Medical Image Analysis, 1(2), 1996.

[2] C. Xu and J. L. Prince, "Generalized gradient vector flow external forces for active contours", Signal Processing - An International Journal, vol. 71, no. 2, pp. 131–139, 1998.

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," Int'l J. Comp. Vis., vol. 1, no. 4, pp. 321–331, 1987.

[4] D. Terzopoulos and K. Fleischer, "Deformable models," The Visual Computer, vol. 4, pp. 306–331, 1988.

[5] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: recovering 3D shape and nonrigid motion," Artificial Intelligence, vol. 36, no. 1, pp. 91– 123, 1988.

[6] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," IEEE Trans. on Computers, vol. 22, no. 1, pp. 67–92, 1973.

[7] B. Widrow, "The "rubber-mask" technique," Pattern Recognition, vol. 5, pp. 175– 211, 1973

[8] A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision," IEEE Trans. Patt. Anal. Mach. Intell., vol. 12, no. 9, pp. 855–867, 1990.

[9] L. D. Cohen, "On active contour models and balloons," CVGIP: Imag. Under., vol. 53, no. 2, pp. 211–218, 1991.

[10] T. McInerney and D. Terzopoulos, "A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis," Comp. Med. Imag. Graph., vol. 19, no. 1, pp. 69–83, 1995.

[11] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours," Numerische Mathematik, vol. 66, pp. 1–31, 1993.

[12] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: a level set approach," IEEE Trans. Patt. Anal. Mach. Intell., vol. 17, no. 2, pp. 158– 175, 1995.

[13] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in Proc. 5th Int'l Conf. Comp. Vis., pp. 694–699, 1995.

[14] R. T. Whitaker, "Volumetric deformable models: active blobs," Tech. Rep. ECRC-94-25, European Computer-Industry Research Centre GmbH, 1994.

[15] G. Sapiro and A. Tannenbaum, "Affine invariant scale-space," Int'l J. Comp. Vis., vol. 11, no. 1, pp. 25–44, 1993

[16] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Shapes, shocks, and deformations I: the components of two-dimensional shape and the reaction-diffusion space," Int'l J. Comp. Vis., vol. 15, pp. 189–224, 1995.

[17] R. Kimmel, A. Amir, and A. M. Bruckstein, "Finding shortest paths on surfaces using level sets propagation," IEEE Trans. Patt. Anal. Mach. Intell., vol. 17, no. 6, pp. 635– 640, 1995

[18] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," Archive for Rational Mechanics and Analysis, vol. 123, no. 3, pp. 199–257, 1993

[19] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," J. Computational Physics, vol. 79, pp. 12–49, 1988.

[20] J. A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science. Cambridge, UK: Cambridge University Press, 2nd ed., 1999.

[21] I. Cohen, L. D. Cohen, and N. Ayache, "Using deformable surfaces to segment 3-D images and infer differential structures," CVGIP: Imag. Under., vol. 56, no. 2, pp. 242–263, 1992.

[22] J. L. Prince and C. Xu, "Nonconservative force models in active geometry," in Proc. IEEE Image and Multidimensional Signal Processing Workshop (IMDSP'98), pp. 139–142, 1998.

[23] R. Ronfard, "Region-based strategies for active contour models," Int'l J. Comp. Vis., vol. 13, no. 2, pp. 229–251, 1994.

[24] C. S. Poon and M. Braun, "Image segmentation by a deformable contour model incorporating region analysis," Phys. Med. Biol., vol. 42, pp. 1833–1841, 1997.

[25] H. Tek and B. B. Kimia, "Volumetric segmentation of medical images by three-dimensional bubbles," Comp. Vis. Imag. Under., vol. 65, pp. 246–258, 1997.

[26] L. D. Cohen and I. Cohen, "Finite- element methods for active contour models and balloons for 2-D and 3-D images," IEEE Trans. Patt. Anal. Mach. Intell., vol. 15, no. 11, pp. 1131–1147, 1993.

[27] P. E. Danielsson, "Euclidean distance mapping," Comp. Graph. Imag. Proc., vol. 14, pp. 227–248, 1980.

[28] G. Borgefors, "Distance transformations in arbitrary dimensions," Comp. Vis. Graph. Imag. Proc., vol. 27, pp. 321–345, 1984.

[29] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," IEEE Trans. Imag. Proc., vol. 7, no. 3, pp. 359–369, 1998.

[30] D. J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," CVGIP: Imag. Under., vol. 55, no. 1, pp. 14–26, 1992.

[31] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformations: deformable superquadrics," IEEE Trans. Patt. Anal. Mach. Intell., vol. 13, pp. 703–714, 1991

[32] A. Gupta, L. von Kurowski, A. Singh, D. Geiger, C. C. Liang, M. Y. Chiu, L. P. Adler, M. Haacke, and D. L. Wilson, "Cardiac MR image segmentation using deformable models," in Proc. IEEE Conf. Computers in Cardiology, pp. 747–750, 1993.

[33] H. Delingette, "Adaptive and deformable models based on simplex meshes," in Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 152–157, 1994.

[34] S. Kumar and D. Goldgof, "Automatic tracking of SPAMM grid and the estimation of deformation parameters from cardiac MR images," IEEE Trans. Med. Imag., vol. 13, pp. 122–132, 1994.

[35] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for de-tecting, tracking, and matching deformable contours," IEEE Trans. Patt. Anal. Mach. Intell., vol. 17, pp. 294–402, 1995.

[36] S. Lobregt and M. A. Viergever, "A discrete dynamic contour model," IEEE Trans. Med. Imag., vol. 14, pp. 12–24, 1995.

[37] C. Nastar and N. Ayache, "Frequency-based nonrigid motion analysis: application to four dimensional medical images," IEEE Trans. Patt. Anal. Mach. Intell., vol. 18, pp. 1067–1079, 1996.

[38] R. Durikovic, K. Kaneda, and H. Yamashita, "Dynamic contour: a texture approach and contour operations," The Visual Computer, vol. 11, pp. 277–289, 1995.

[39] T. McInerney and D. Terzopoulos, "Topologically adaptable snakes," in Proc. 5th Int'l Conf. Comp. Vis., pp. 840–845, 1995

[40] B. B. Kimia, Conservation Laws and a Theory of Shape. Ph.D. thesis, McGill Centre for Intelligent Machines, McGill University, Montreal, Canada, 1990.

[41] M. A. Grayson, "Shortening embedded curves," Annals of Mathematics, vol. 129, pp. 71–111, 1989.

[42] J. A. Sethian, "Curvature and evolution of fronts," Commun. Math. Phys., vol. 101, pp. 487–499, 1985.

[43] J. A. Sethian, "A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed," J. Differential Geometry, vol. 31, pp. 131–161, 1989.

[44] G. Sapiro, "Geometric partial differential equations in image analysis: past, present, and future," in Proc. IEEE Int'l Conf. Imag. Proc., vol. 3, pp. 1–4, 1995.

[45] D. Adalsteinsson and J. A. Sethian, "The fast construction of extension velocities in level set methods," J. Computational Physics, vol. 148, pp. 2–22, 1999

[46] J. A. Sethian, An Analysis of Flame Propagation, Ph.D. thesis, Dept. of Mathematics, University of California, Berkeley, CA, 1982.

[47] L. H. Staib and J. S. Duncan, "Boundary finding with parametrically deformable models," IEEE Trans. Patt. Anal. Mach. Intell., vol. 14, no. 11, pp. 1061–1075, 1992.

[48] K. Delibasis, P. E. Undrill, and G. G. Cameron, "Designing Fourier descriptor-based geometric models for object interpretation in medical images using genetic algorithms," Comp. Vis. Imag. Under., vol. 66, pp. 286–300, 1997.

[49] A. Pentland and B. Horowitz, "Recovery of nonrigid motion and structure," IEEE Trans. Patt. Anal. Mach. Intell., vol. 13, pp. 730–742, 1991.

[50] K. H. Huebner, E. A. Thornton, and T. G. Byrom, The Finite Element Method for Engineers. New York: John Wiley & Sons, 3rd ed., 1994.

[51] E. Bardinet, L. D. Cohen, and N. Ayache, "A parametric deformable model to fit unstructured 3D data," Comp. Vis. Imag. Under., vol. 71, pp. 39–54, 1998.

[52] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "Use of active shape models for locating structures in medical images," Imag. Vis. Computing J., vol. 12, no. 6, pp. 355–366, 1994.

[53] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models – their training and application," Comp. Vis. Imag. Under., vol. 61, no. 1, pp. 38–59, 1995.

[54] J. C. Gower, "Generalized Procrustes analysis," Psychometrika, vol. 40, pp. 33–51, 1975.

[55] N. Duta and M. Sonka, "Segmentation and interpretation of MR brain images: an improved active shape model," IEEE Trans. Med. Imag., vol. 17, pp. 1049–1062, 1998.

[56] Y. Wang and L. H. Staib, "Boundary finding with correspondence using statistical shape models," in Proc. IEEE Conf. Comp. Vis. Patt. Recog., pp. 338–345, 1998

[57] H. H. S. Ip and D. Shen, "An affine-invariant active contour model (AI-snake) for model-based segmentation," Imag. Vis. Computing J., vol. 16, pp. 135–146, 1998.

[58] Y. Guo and B. C. Vemuri, "Hybrid geometric active models for shape recovery in medical images," in Proc. Information Processing in Medical Imaging (IPMI'99), pp. 112–125, 1999.

[59] D. Fritsch, S. Pizer, L. Yu, V. Johnson, and E. Chaney, "Segmentation of medical im-age objects using deformable shape loci," in Proc. Information Processing in Medical Imaging (IPMI'97), pp. 127–140, 1997.

[60] Chenyang Xu, D. L. Pham, J. L. Prince, "Image Segmentation using Deformable Models".

[61] Scott. Acton, Nilanjan Ray, "Biomedical Image Analysis: Segmentation", Synthesis Lectures on Image, Video and Multimedia Processing, #9, Morgan & Claypool Publishers.

[62] C. Xu, D. L. Pham, and J. L. Prince, "Medical Image Segmentation Using Deformable Models," Handbook of Medical Imaging, Volume 2: Medical Image Processing and Analysis, pp. 129-174, edited by J.M. Fitzpatrick and M. Sonka, SPIE Press, May 20.