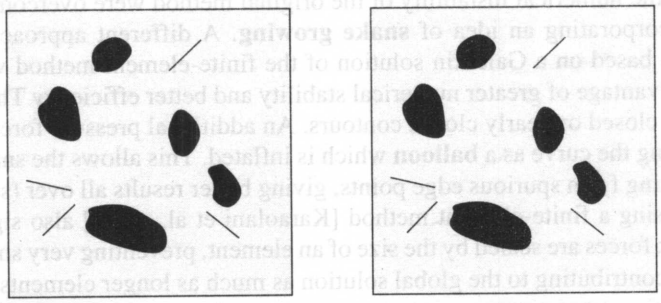
$$E_{\text{term}} = \frac{\partial \psi}{\partial \mathbf{n}_R} = \frac{\partial^2 g / \partial \mathbf{n}_R^2}{\partial g / \partial \mathbf{n}}$$

$$= \frac{\left(\partial^2 g / \partial y^2\right)\left(\partial g / \partial x\right)^2 - 2\left(\partial^2 g / \partial x \partial y\right)\left(\partial g / \partial x\right)\left(\partial g / \partial y\right) + \left(\partial^2 g / \partial x^2\right)\left(\partial g / \partial y\right)^2}{\left(\left(\partial g / \partial x\right)^2 + \left(\partial g / \partial y\right)^2\right)^{3/2}}. \tag{7.20}$$

The snake behavior may be controlled by adjusting the weights ω_{line}, ω_{edge}, wterm. A snake attracted to edges and terminations is shown in Figure 7.8.



(a) (b)

Figure 7.8: A snake attracted to edges and terminations. (a) Contour illusion. (b) A snake attracted to the subjective contour. Adapted from [Kass et al., 1987b].

The third term of the integral (7.15) comes from external constraints imposed either by a user or some other higher-level process which may force the snake toward or away from particular features. If the snake is near to some desirable feature, the energy minimization will pull the snake the rest of the way. However, if the snake settles in a local energy minimum that a higher-level process determines as incorrect, an area of energy peak may be made at this location to force the snake away to a different local minimum.

A contour is defined to lie in the position in which the snake reaches a local energy minimum. From equation (7.15), the functional to be minimized is

$$E^*_{\text{snake}} = \int_0^1 E_{\text{snake}}\left(\mathbf{v}(s)\right) \mathrm{d}s.$$

Then, from the calculus of variations, the Euler-Lagrange condition states that the spline $\mathbf{v}(s)$ which minimizes $E*_{\text{snake}}$ must satisfy

$$\frac{\mathrm{d}}{\mathrm{d}s} E_{\mathbf{v}_s} - E_{\mathbf{v}} = 0, \tag{7.21}$$

where $E_{\mathbf{v}_s}$ is the partial derivative of E with respect to $\mathrm{d}\mathbf{v}/\mathrm{d}s$ and $E_{\mathbf{v}}$ is the partial derivative of E with respect to $\mathbf{v}$. Using equation (7.16) and denoting $E_{\text{ext}} = E_{\text{image}} + E_{\text{con}}$, the previous equation reduces to

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\alpha(s)\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}s}\right) + \frac{\mathrm{d}^2}{\mathrm{d}s2}\left(\beta(s)\frac{\mathrm{d}^2\mathbf{v}}{\mathrm{d}s^2}\right) + \nabla E_{\text{ext}}\left(\mathbf{v}(s)\right) = 0 \tag{7.22}$$

To solve the Euler-Lagrange equation, suppose an initial estimate of the solution is available. An evolution equation is formed:

$$\frac{\partial \mathbf{v}(s,t)}{\partial t} - \frac{\partial}{\partial s}\left(\alpha(s)\frac{\partial \mathbf{v}(s,t)}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(\beta(s)\frac{\partial^2 \mathbf{v}(s,t)}{\partial s^2}\right) + \nabla E_{\text{ext}}\left(\mathbf{v}(s,t)\right) = 0. \tag{7.23}$$

The solution is found if $\partial \mathbf{v}(s, t)/\partial t = 0$. Nevertheless, minimization of the snake energy integral is still problematic; numerous parameters must be designed (weighting factors, iteration steps, etc.), a reasonable initialization must be available, and, moreover, the solution of the Euler-Lagrange equation suffers from numerical instability.

Originally, a resolution minimization method was proposed [Kass et al., 1987a]; partial derivatives in s and t were estimated by the finite-differences method. Later [Amini et al., 1988, 1990], a dynamic programming approach was proposed which allows 'hard' constraints to be added to the snake. Further, a requirement that the internal snake energy must be a continuous function may thus be eliminated and some snake configurations may be prohibited (that is, have infinite energy), allowing more a priori knowledge to be incorporated.

Difficulties with the numerical instability of the original method were overcome by Berger [Berger and Mohr, 1990] by incorporating an idea of **snake growing**. A different approach to the energy integral minimization that is based on a Galerkin solution of the finite-element method was proposed in [Cohen, 1991] and has the advantage of greater numerical stability and better efficiency This approach is especially useful in the case of closed or nearly closed contours. An additional pressure force is added to the contour interior by considering the curve as a **balloon** which is inflated. This allows the snake to overcome isolated energy valleys resulting from spurious edge points, giving better results all over (see Figures 7.9 and 7.10). Another approach using a finite-element method [Karaolani et al., 1992] also significantly improves the solution's efficiency; forces are scaled by the size of an element, preventing very small contributions (which may be noise) from contributing to the global solution as much as longer elements.

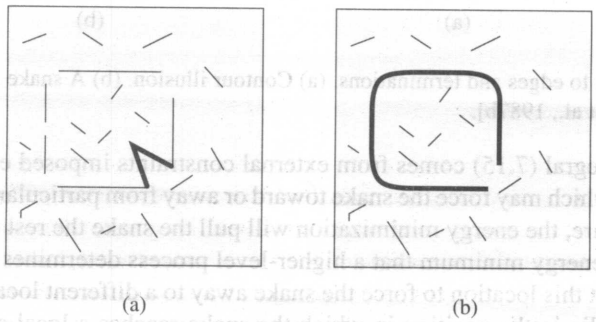


Figure 7.9: Active contour model—balloon. (a) Initial contour. (b) Final contour after inflation and energy minimization. Adapted from [Cohen and Cohen, 1992].

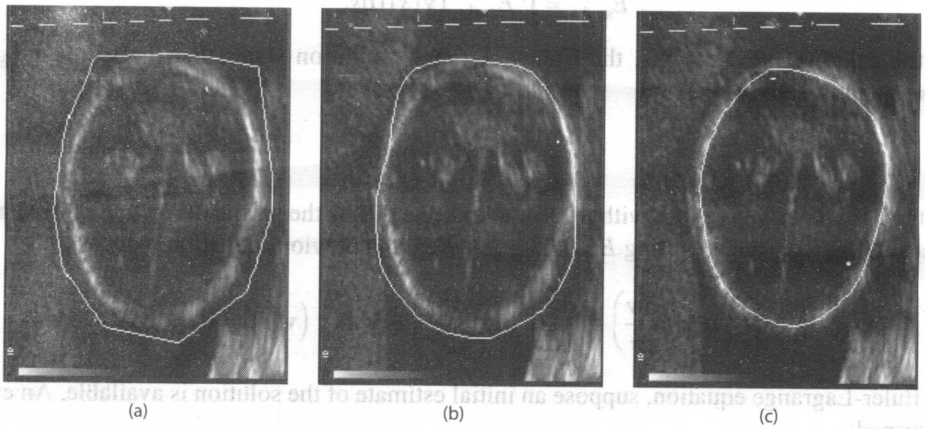


Figure 7.10: Balloon-based image segmentation of an ultrasound image of a fetal head. (a) Initial position of the balloon. (b) Balloon deformation after 10 iterations. (c) Final position of the balloon after 25 iterations. *Courtesy of V. Chalana.*

7.2.2 Extensions

Deformable models based on active contours were generalized to three dimensions by Terzopoulos [Terzopoulos et al., 1987, 1988; McInerney and Terzopoulos, 1993], and 3D balloons were introduced in [Cohen and Cohen, 1992]; more recently [Bulpitt and Efford, 1996], a 3D generalization using an adaptive mesh that can refine and decimate has been successfully applied to modeling MR images of human heads and hands. In [Zhang and Braun, 1997], a fully three-dimensional active surface model with self-inflation and self-deflation forces has been introduced. Further, fast algorithms for active contour models [Williams and Shah, 1992; Olstad and Tysdahl, 1993; Lam and Yan, 1994] and a number of other extensions exist. Several problems often accompany the snake-based approach: Snakes tend to be attracted to spurious edges, they sometimes degenerate in shape by shrinking and flattening, and the convergence and stability of the contour deformation process may be unpredictable. Additionally, while their initialization is not straightforward, it may affect their performance substantially. Many approaches have been designed to overcome these limitations, some of them using more complex knowledge and models, employing neural networks, adaptively modifying the number of control points in snakes represented by B-splines [Figueiredo et al., 1997], designing internal forces independent of contour shape, determining external forces from edge images [Xu and Prince, 1998], limiting the region of interest by a pair of snakes, simplifying the initialization process [Neuenschwander et al., 1994], considering region-based information [Etoh et al., 1993; Ronfard, 1994], and allowing topological snake adaptation so that the snake can flow into complex shapes including branches [McInerney and Terzopoulos, 1995].

Over the years, several variants of active contour models emerged as improvements of the original **finite difference** method discussed above. These variants aimed at an increase of robustness against noise, decrease of initialization sensitivity, improved selectivity for certain classes of objects, etc. The main such improvements include **finite element snakes** [Cohen and Cohen, 1993], **B-snakes** [Menet et al., 1990; Blake and Isard, 1998], and **Fourier deformable models** [Staib and Duncan, 1992]. None of them has emerged as a gold standard. Recently, a snake unification approach appeared called the **united snakes** combining several snake variants—namely the finite difference, B-spline, and **Hermite polynomial** snakes in a comprehensive finite element formulation [Liang et al., 1999, 2006]. The *united snakes* also include unification with live wire and intelligent scissor methods for interactive contour-based segmentation (Section 6.2.5). Detailed description of the *united snakes* theory and framework can be found in [Liang et al., 2006].

7.2.3 Gradient vector flow snakes

Two main limitations common to these approaches are the requirement of snake initialization being close to the desired solution, and difficulties in segmenting concave portions of the boundary. To overcome these problems, **gradient vector flow** (GVF) fields and their use in snake image segmentation were reported in [Xu and Prince, 1998].

GVF field is a non-irrotational external force field that points toward the boundaries when in their proximity and varies smoothly over homogeneous image regions all the way to image borders. Consequently, it can drive a snake toward a border from a large distance and can segment object concavities. In comparison to the classical snake approach [Kass et al., 1987a] in which similar behavior can be attempted (but not obtained) by blurring the edge image or using pressure forces, it does not suffer from edge localization problems caused by edge distortion from their smoothing, it does not require to carefully fine-tune the balloon pressure forces to overcome the noise but not overcome the salient image features, and can attract the snake when initialized at either side of the boundary.

The GVF field is derived from an image by minimization of an energy functional by solving decoupled linear partial differential equations via diffusing the gradient vectors of the edge image. The GVF is then used as an external force in the snake equations (7.15), (7.22) forming a **GVF snake**. Considerable insensitivity to initialization and ability to segment concave boundaries result. The GVF field $\mathbf{g}(x, y) = (u(x, y), v(x, y))$ minimizes the energy functional

$$E = \iint \mu\left(u_x^2 + y_y^2 + v_x^2 + v_y^2\right) + |\nabla f|^2 |\mathbf{g} - \nabla f|^2 \, dx \, dy, \tag{7.24}$$

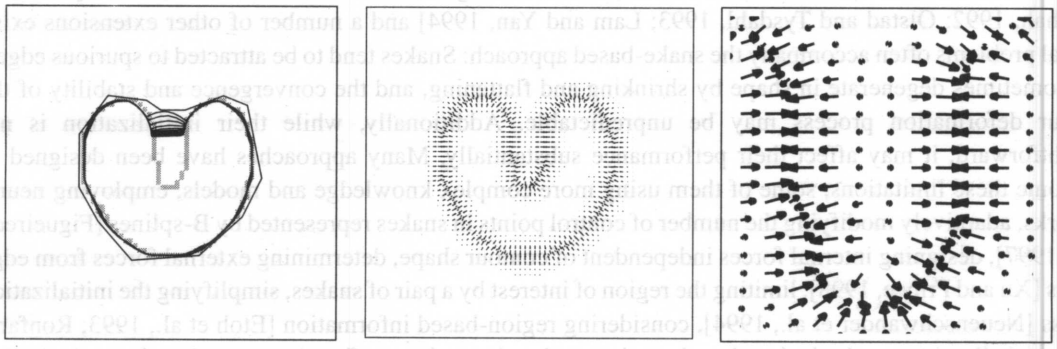


Figure 7.11: Classic snake convergence. (a) Convergent sequence of snake locations. Note that the snake fails segmenting the concave boundary. (b) Classic external forces. (c) Close-up of the concave object region. No forces exist capable of pulling the snake inside the bay. *Courtesy of J. L. Prince and C. Xu, Johns Hopkins University, ©1998 IEEE [Xu and Prince, 1998].*

where μ is a regularization parameter balancing the weight of the first and second terms (increasing μ with increased noise), the subscripts denote directional partial derivatives. As described in [Xu and Prince, 1998], the GVF can be obtained by solving the Euler equations

$$\mu \nabla^2 u - \{u - f_x\}\{f_x^2 + f_y^2\} = 0, \tag{7.25}$$

$$\mu \nabla^2 v - (v - f_y (f_x^2 + f_y^2) = 0, \tag{7.26}$$

where ∇^2 is a Laplacian operator. The second term in equations (7.25), (7.26) is zero in homogeneous regions since the directional gradients f_x, f_y are zero. Consequently, the GVF behavior in the homogeneous regions is fully defined by the Laplace equation effectively diffusing the information from the boundaries to the homogeneous parts of the image. Solutions to equations (7.25), (7.26) can be found by treating u and v as functions of time and solving the following two decoupled equations for $t \to \infty$:

$$u_t(x, y, t) = \mu \nabla^2 u(x, y, t) - (u(x, y, t) - f_x(x, y))(f_x(x, y)^2 + f_y(x, y)^2), \tag{7.27}$$

$$v_t(x, y, t) = \mu \nabla^2 v(x, y, t) - (v(x, y, t) - f_y(x, y))[f_x(x, y)^2 + f_y(x, y)^2). \tag{7.28}$$

These **generalized diffusion equations** can be solved as separate scalar partial differential equations in u and v and are used in heat conduction, fluid flow, etc. [Charles and Porsching, 1990].

Once $\mathbf{g}(x, y)$ is computed, equation (7.22) is modified using the GVF external force $E_{\text{ext}} = \mathbf{g}(x, y)$ yielding the GVF snake equation

$$\mathbf{v}_t(s, t) = \alpha \mathbf{v}''(s, t) - \beta \mathbf{v}''''(s, t) + \mathbf{g}, \tag{7.29}$$

which can be solved as the traditional snake equation by an iterative process following discretization.

Figure 7.11 shows a convergence process and snake-attracting forces of a classic snake. It can be seen that there is no force that would pull the snake towards the concave portion of the boundary. Consequently,

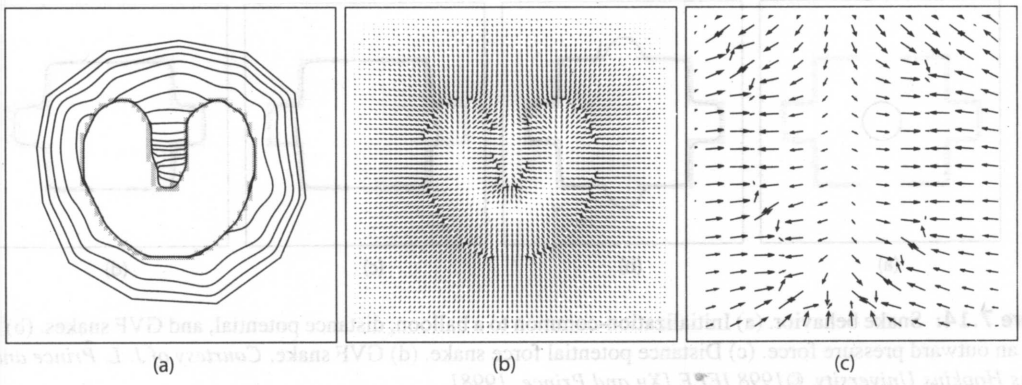


Figure 7.12: GVF snake convergence. (a) Convergent sequence of snake locations. Note that the snake succeeded in segmenting the concave boundary. (b) GVF external forces. (c) Close-up of the concave object region. Forces exist capable of pulling the snake inside the bay. *Courtesy of J. L. Prince and C. Xu, Johns Hopkins University, ©1998 IEEE [Xu and Prince, 1998].*

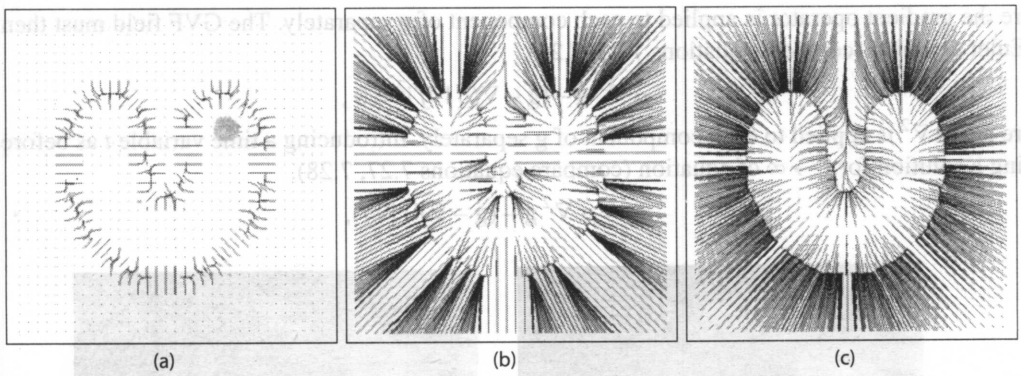


Figure 7.13: Streamlines originating in a regular 32 × 32 grid of points. (a) Classic potential force field–only locations very close to the border can be attracted to the object. (b) Distance-based external force field. Note that there is insufficient pull of the locations inside the bay to correctly segment the concave region. However, the snake can be initialized at a distance from the object. (c) GVF force field demonstrating the ability to correctly segment the concave region and maintaining the ability of a large-distance initialization. *Courtesy of J. L. Prince and C. Xu, Johns Hopkins University, ©1998 IEEE [Xu and Prince, 1998].*

the segmentation fails in that area. When adding distance-based forces, the snake segmentation fails in a similar fashion. In comparison, the GVF snake successfully segments the object as demonstrated in Figure 7.12. Figure 7.13 shows the attraction force coverage of an entire image using the classic, distance-based, and GVF field forces clearly demonstrating the advantages of the GVF approach. Figure 7.14 compares the segmentation of an object with incomplete, concave, and convex boundary segments when initialized from within the object. Importantly, the GVF snake gives virtually the same segmentation if initialized from the outside with no change of the parameters. Figure 7.15 shows an application of a GVF snake to cardiac MR segmentation.

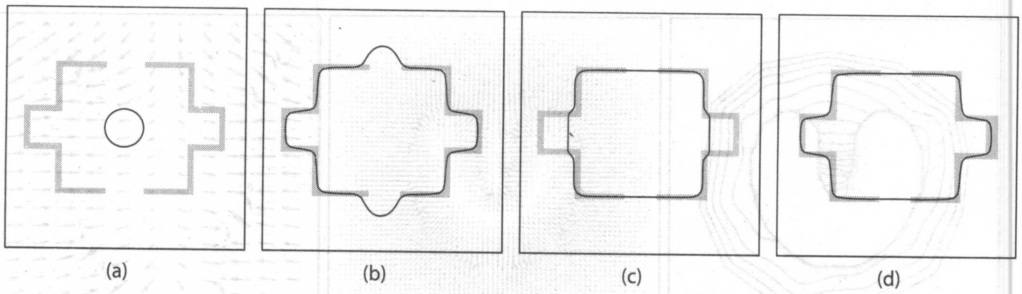


Figure 7.14: Snake behavior. (a) Initialization-common to a balloon, distance potential, and GVF snakes. (b) Balloon with an outward pressure force. (c) Distance potential force snake. (d) GVF snake. *Courtesy of J. L. Prince and C. Xu, Johns Hopkins University, ©1998 IEEE [Xu and Prince, 1998].*

GVF can be generalized to higher dimensions defining the d-dimensional GVF field $\mathbf{g}(\mathbf{x})$ as minimizing the energy functional (compare equation 7.24).

$$E = \int_{\mathcal{R}^d} \mu |\nabla \mathbf{g}|^2 + |\nabla f|^2 |\mathbf{g} - \nabla f|^2 \, \mathrm{d}\mathbf{x} \tag{7.30}$$

where the gradient operator is applied to each component of $\mathbf{g}$ separately. The GVF field must then satisfy the Euler equation (compare equations 7.25, 7.26).

$$\mu \nabla^2 \mathbf{g} - (\mathbf{g} - \nabla f) |\nabla f|^2 = 0 \,, \tag{7.31}$$

where again ∇^2 is applied to each component of $\mathbf{g}$ separately. Introducing a time variable t as before allows finding a solution for $t \to \infty$ of equation (compare equations 7.27, 7.28).

$$\mathbf{g}t = \mu \nabla^2 \mathbf{g} - (\mathbf{g} - \nabla f) |\nabla f|, \tag{7.32}$$

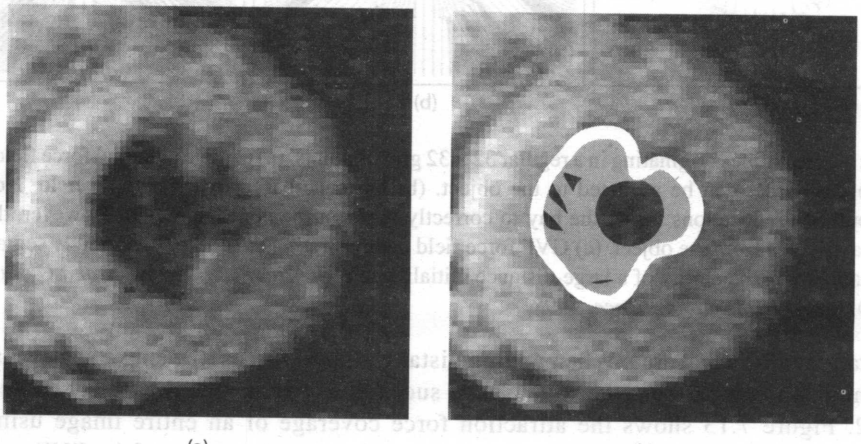


Figure 7.15: Cardiac MR segmentation using GVF snakes. (a) Original short axis MR image of the left cardiac ventricle. (b) GVF snake segmentation showing the convergence process. *Courtesy of J. L. Prince and C. Xu, Johns Hopkins University, ©1998 IEEE [Xu and Prince, 1998].*

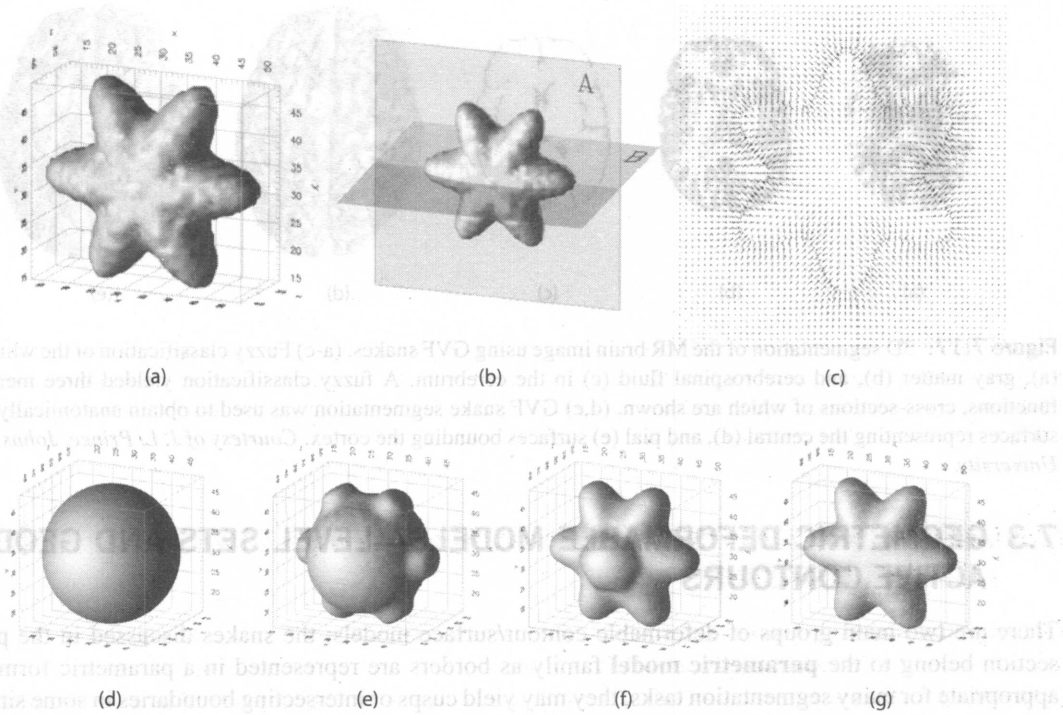


Figure 7.16: GVF snake segmentation in 3D. (a) Isosurface of a 3D object defined on a 64^3 grid. (b) Position of plane A on which the 3-D GVF vectors are depicted in (c). (d) The initial configuration of a deformable surface using GVF and its positions after (e) 10, (f) 40, and (g) 100 iterations. *Courtesy of J. L. Prince and C. Xu, Johns Hopkins University, ©1998 IEEE [Xu and Prince, 1998].*

where $\mathbf{g}_t$ denotes a partial derivative with respect to t. Similarly to the 2D case, equation (7.32) contains d decoupled scalar linear second order parabolic partial differential equations in each element of $\mathbf{g}$ and can be solved iteratively. Figure 7.16 demonstrates a GVF snake segmentation of a star-shaped object in 3D. Figure 7.17 shows an application of a 3D GVF snake in brain segmentation [Tosun et al., 2004].

Active contour models represent a recent approach to contour detection and image interpretation. They differ substantially from classical approaches, where features are extracted from an image and higher-level processes try to interpolate sparse data to find a representation that matches the original data—active contour models start from an initial estimate based on higher-level knowledge, and an optimization method is used to refine the initial estimate. During the optimization, image data, an initial estimate, desired contour properties, and knowledge-based constraints are considered. Feature extraction and knowledge-based constrained grouping of these features are integrated into a single process, which seems to be the biggest advantage. Active contour models, however, search for *local* energy minima not attempting to achieve globally optimal solutions. Applications can be found in many areas of machine vision and medical image analysis [Cohen and Cohen, 1992; Hyche et al., 1992; Lobregt and Viergever, 1995].

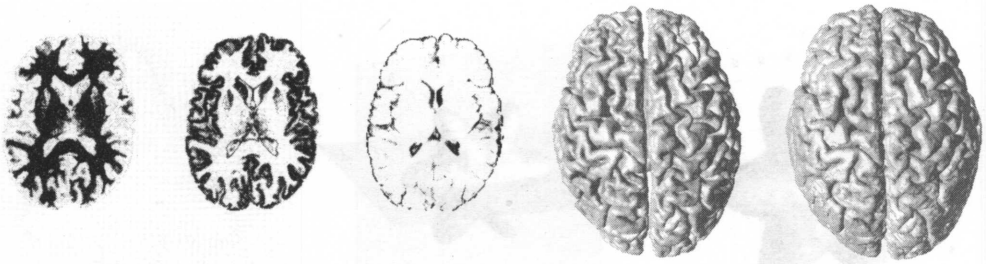


Figure 7.17: 3D segmentation of the MR brain image using GVF snakes. (a-c) Fuzzy classification of the white matter (a), gray matter (b), and cerebrospinal fluid (c) in the cerebrum. A fuzzy classification yielded three membership functions, cross-sections of which are shown. (d,e) GVF snake segmentation was used to obtain anatomically feasible surfaces representing the central (d), and pial (e) surfaces bounding the cortex. *Courtesy of J. L. Prince, Johns Hopkins University.*

7.3 GEOMETRIC DEFORMABLE MODELS—LEVEL SETS AND GEODESIC ACTIVE CONTOURS

There are two main groups of deformable contour/surface models: the snakes discussed in the previous section belong to the **parametric model** family as borders are represented in a parametric form. While appropriate for many segmentation tasks, they may yield cusps or intersecting boundaries in some situations. The second family of deformable surfaces—**geometric deformable models**—overcome this problem by representing developing surfaces by partial differential equations. The geometric deformable model literature has grown extensively in the past years, in many cases reporting a variety of applications in which deformable model based segmentation can be used. An excellent treatment of deformable model principles and their comparisons can be found in [Xu et al., 2000], which also provided conceptual guidance to this section.

Geometric deformable models were introduced independently by Malladi et al. and Caselles et al. and named **level set front propagation** and **geodesic active contour** segmentation approaches [Caselles et al., 1993; Malladi et al., 1993, 1995]. The main feature separating geometric deformable models from parametric ones is that curves are evolved using only geometric computations, independent of any parameterization: the process is *implicit*. Consequently, the curves and/or surfaces can be represented as *level sets* of higher dimensional functions yielding seamless treatment of topological changes. Hence, without resorting to dedicated contour tracking, unknown numbers of multiple objects can be detected simultaneously. Detailed treatment of curve evolution theory and level set methods can be found in [Osher and Sethian, 1988; Sethian, 1999; Sapiro and Tannenbaum, 1993; Kimia et al., 1995; Alvarez et al., 1993; Osher and Fedkiw, 2002; Osher and Paragios, 2003].

Let a curve moving in time t be denoted by $\mathbf{X}(s, t) = [X(s, t), Y(s, t)]$, where s is curve parameterization. Let $\mathbf{N}$ be the moving curve's inward normal, and c curvature, and let the curve develop along its normal direction according to the partial differential equation

$$\frac{\partial \mathbf{X}}{\partial t} = V(c)\,\mathbf{N}. \tag{7.33}$$

Here, curve evolution is defined by the **speed function** $V(c)$: Figure 7.18 demonstrates the concept of front evolution. As the curve is moving, it may need to be reparameterized to satisfy equation (7.33).

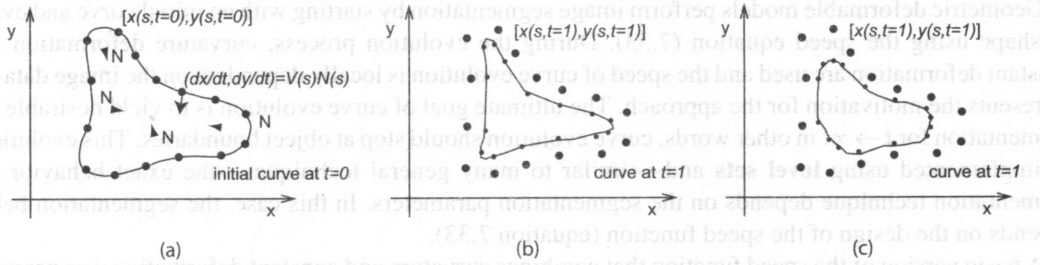


Figure 7.18: Concept of front evolution. (a) Initial curve at $t = 0$. (b) Curve at $t = 1$. Note that each curve point moved in direction of **N** by distance given by velocity V. (c) Curve at $t = 1$ assuming the velocity $V(c)$ is a function of curvature.

If the curve evolution is driven by a **curvature deformation** equation, the partial differential equation describes a curve smoothing process which removes potential singularities and eventually shrinks the curve to a point

$$\frac{\partial \mathbf{X}}{\partial t} = \alpha c \mathbf{N}, \tag{7.34}$$

where α is a constant—similar to elastic internal forces used in snakes (Section 7.2). Figure 7.19 shows deformation behavior using positive ($\alpha > 0$) and negative ($\alpha < 0$) curvature.

Curve deformation driven by the **constant deformation** equation (7.35) is complementary, and is similar to the inflation balloon force discussed earlier (Section 7.2) and may introduce singularities like sharp corners

$$\frac{\partial \mathbf{X}}{\partial t} = V_0 \mathbf{N}, \tag{7.35}$$

where V_0 determines constant speed of deformation,

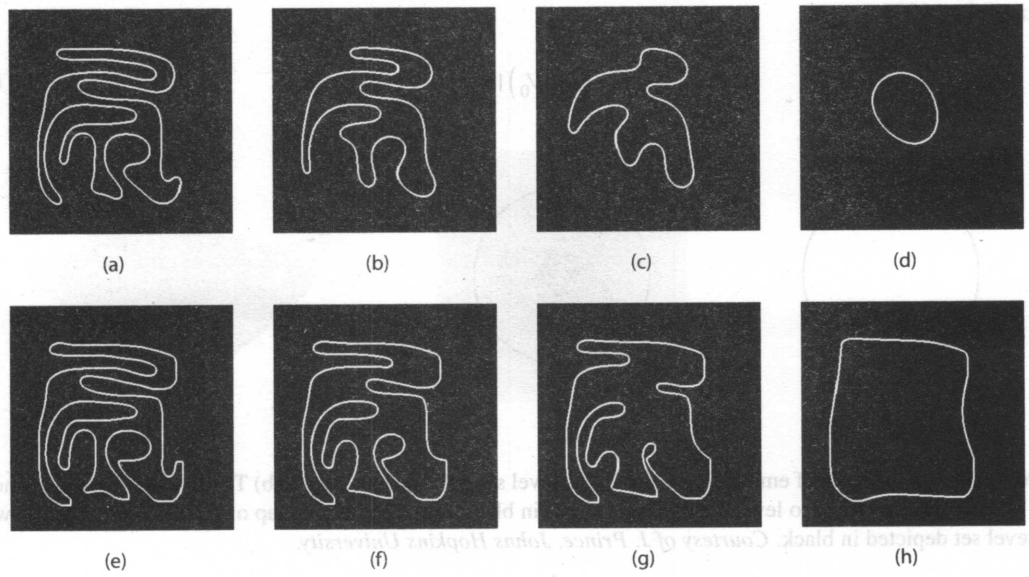


Figure 7.19: Evolution of a closed 2D curve using curvature deformation. (a-d) Using positive curvature, iterations 100, 2,000, 4,000, 17,000. (e-h) Using negative curvature, iterations 100, 2,000, 4,000, 17,000.

Geometric deformable models perform image segmentation by starting with an initial curve and evolving its shape using the speed equation (7.33). During the evolution process, curvature deformation and/or constant deformation are used and the speed of curve evolution is locally dependent on the image data— this represents the motivation for the approach. The ultimate goal of curve evolution is to yield desirable image segmentation for $t \to \infty$: in other words, curve evolution should stop at object boundaries. This evolution can be implemented using **level sets** and—similar to many general techniques—the exact behavior of the segmentation technique depends on the segmentation parameters. In this case, the segmentation behavior depends on the design of the speed function (equation 7.33).

A basic version of the speed function that combines curvature and constant deformation was proposed in [Caselles et al., 1993; Malladi et al., 1995], and is of the form

$$\frac{\partial \phi}{\partial t} = k\left(c + V_0\right)|\nabla \phi|, \tag{7.36}$$

where

$$k = \frac{1}{1 + |\nabla(G_\sigma * I)|}. \tag{7.37}$$

Here, ϕ represents the propagating curve front (and denotes a level set function, see below). The term $\nabla(G_\sigma * I)$ denotes gradient of a Gaussian-smoothed image, where σ is a smoothing parameter. As can be seen, a positive value of V_0 expands the curve while k serves as a stopping term—with $k \to 0$ for image locations exhibiting a large image gradient, i.e., image edges. Clearly, the edges must be strong for the curve evolution to stop (or rather, almost stop; a simple edge-strength threshold may be used to force a slow-moving front to truly stop). An obvious problem with this speed function is that it will not slow down sufficiently at weaker or indistinct boundaries, and once the curve passes the boundary location, it will continue moving with no force pulling it back. Figure 7.21 gives an example of segmentation sensitivity to the stopping criterion.

An energy minimization approach to overcome this behavior was introduced in [Caselles et al., 1997; Yezzi et al., 1997]

$$\frac{\partial \phi}{\partial t} = k\left(c + V_0\right)|\nabla \phi| + \nabla k \, \nabla \phi. \tag{7.38}$$

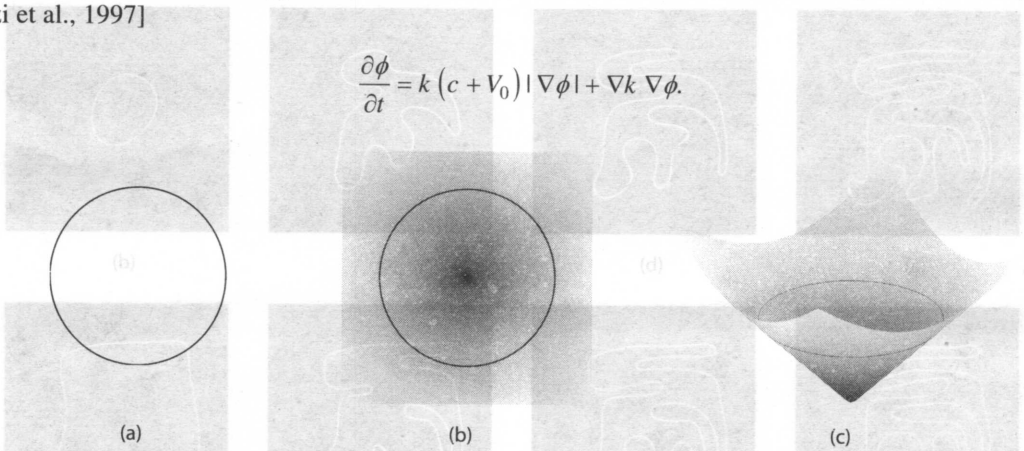


Figure 7.20: An example of embedding a curve as a level set. (a) A single curve. (b) The level set function where the curve is embedded as the zero level set $\phi[\mathbf{X}(s, t), t] = 0$ (in black). (c) The height map of the level set function with its zero level set depicted in black. *Courtesy of J. Prince, Johns Hopkins University.*

The additional stopping term $\nabla k \nabla \phi$ acts to pull the boundary back to the object border once the curve evolution passes it. Other speed functions can be found in [Siddiqi et al., 1998].

An important question however remains—how to efficiently execute the curve evolution process. The idea that made geometric deformable models feasible is to represent the segmentation boundary/surface

implicitly as a level set of a higher-dimensional function— the **level set function** ϕ—defined on the same image domain [Osher and Sethian, 1988; Sethian, 1985, 1989]. Using the level set representation of the curve allows its evolution by updating the level set function $\phi(t)$ at fixed time points. Instances of curve evolution are obtained by determination of the zero-level set for individual time points ($\phi(t) = 0$). In other words, the evolving curve at time t is found as a set of points on the image domain for which the function value (height) of the level set function at time t is equal to zero, and the final solution is given by the zero-level set $\phi(t - \infty) = 0$. Importantly, the level set function remains a valid function during the updating process even if the embedded level set curve may change topology, develop singularities, etc. Figures 7.20 and 7.22 illustrate the level set concept of curve embedding, its evolution, and topology change.

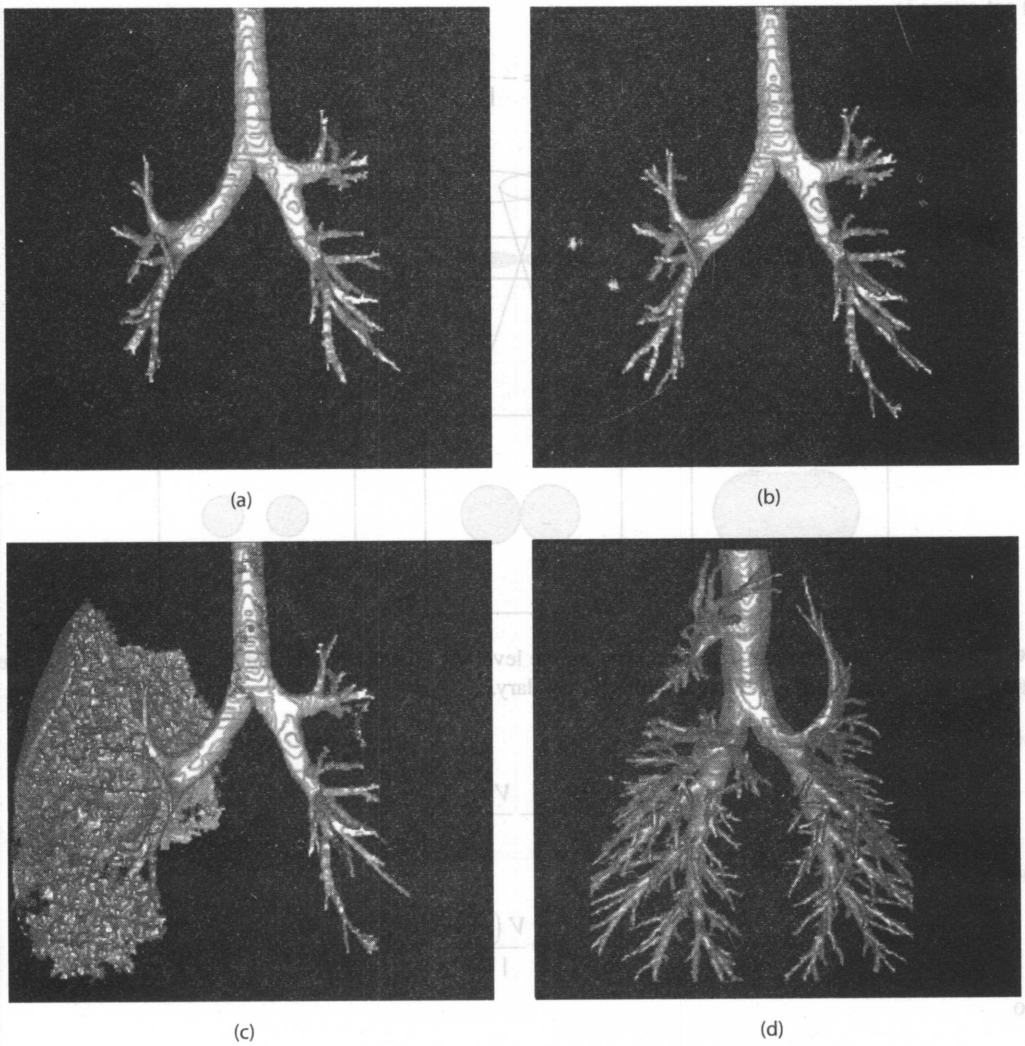


Figure 7.21: Pulmonary airway tree segmentation using a 3D fast marching level set approach applied to X-ray computed tomography data. The speed function was defined as $V = 1/intensity$; the segmentation front moves faster in dark regions corresponding to air in CT images and slower in bright regions corresponding to airway walls. The stopping criterion uses a combination of gradient threshold value T_g and local image intensity threshold T_i—increasing gradient and image intensity slow down and stop the front propagation. (a) Human airway tree segmentation result employing $T_i = 6$. (b) $T_i = 11$. (c) $T_i = 13$—a segmentation leak occurs. (d) Sheep airway tree segmentation—obtaining a larger number of airways is due to a higher X-ray dose yielding better quality image data.

A more formal treatment to define a level set embedding of the curve evolution equation (7.33) is now appropriate. Having a level set function $\phi(x, y, t)$ with the contour X (s, t) as its zero-level set, the situation is described by

$$\phi(\mathbf{X}(s, t), t) = 0. \tag{7.39}$$

If this equation is differentiated with respect to t and the chain rule is used,

$$\frac{\partial \phi}{\partial t} + \nabla\phi \frac{\partial \mathbf{X}}{\partial t} = 0 \tag{7.40}$$

Now assuming that ϕ is negative inside the zero-level set and positive outside, the inward unit normal to the level set curve is

$$\mathbf{N} = -\frac{\nabla\phi}{|\nabla\phi|}$$

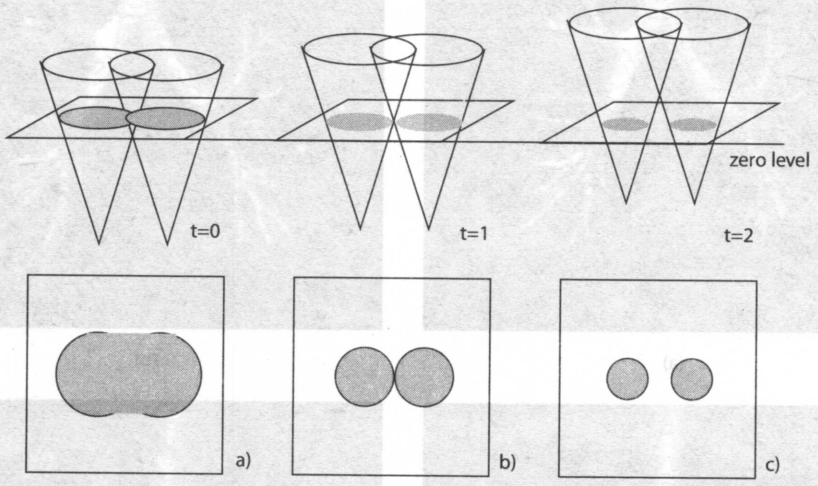


Figure 7.22: Topology change using level sets. As the level set function is updated for t = 1,2, 3, the zero-level set changes topology, eventually providing a 2-object boundary.

and so from the speed equation (7.33)

$$\frac{\partial \mathbf{X}}{\partial t} = -\frac{V(c)\nabla\phi}{|\nabla\phi|} \tag{7.41}$$

and hence

$$\frac{\partial \phi}{\partial t} - \nabla\phi \frac{V(c)\nabla\phi}{|\nabla\phi|} = 0 \tag{7.42}$$

and so

$$\frac{\partial \phi}{\partial t} = V(c)|\nabla\phi|. \tag{7.43}$$

The curvature c at the zero-level set is

$$c = \nabla \frac{\nabla\phi}{|\nabla\phi|} = \frac{\phi_{xx}\,\phi_y^2 - 2\,\phi_x\,\phi_y\,\phi_{xy} + \phi_{yy}\,\phi_x^2}{\left(\phi_x^2 + \phi_y^2\right)^{3/2}}. \tag{7.44}$$

Equation (7.43) shows how to perform curve evolution specified by equation (7.33) using the level set method.

To implement geometric deformable contours, an initial level set function $\phi(x, y, t = 0)$ must be defined, the speed function must be derived for the entire image domain, and evolution must be defined for locations in which normals do not exist due to the development of singularities. The initial level set function is frequently based on the signed distance $D(x, y)$ from each grid point to the zero-level set, $\phi(x, y, 0) = D(x, y)$. An efficient algorithm for construction of the signed distance function is called a **fast marching method** [Malladi and Sethian, 1996, 1998; Sethian, 1999].

Note that the evolution equation (7.43) is only derived for the zero-level set. Consequently, the speed function $V(c)$ is undefined on other level sets and needs to be extended to all level sets. A number of extension approaches, including the frequently used **narrow band** extension, can be found in [Malladi et al., 1995; Sethian, 1999]. Although the equations for $\mathbf{N}$ and c hold for all level sets, the distance function property may become invalid over the course of curve evolution causing inaccuracies in curvature and normal vector computations. Consequently, reinitialization of the level set function to a signed distance function is often required. Another method [Adalsteinsson and Sethian, 1999] does not suffer from this problem.

As described above, using the *constant deformation* approach may cause sharp corners of the zero-level set resulting in an ambiguous normal direction. In that case, the deformation can be continued using an **entropy condition** [Sethian, 1982].

The speed function given in equation (7.36) uses the image gradient to stop the curve evolution. To overcome the inherent problems of edge-based stopping criteria, considering the region properties of the segmented objects is frequently helpful. For example, a piecewise constant minimal variance criterion based on the Mumford-Shah functional [Mumford and Shah, 1989] was proposed by Chan and Vese [Chan and Vese, 2001] to deal with such situations. Considering a 2D image consisting of pixels $I(x, y)$ and the segmentation defined by an evolving closed zero-level set curve ϕ, the **Chan–Vese energy functional** is

$$\begin{aligned} C(\phi, a_1, a_2) &= C_1(\phi, a_1, a_2) + C_2(\phi, a_1, a_2) \\ &= \int_{inside(\phi)} \left(I(x, y) - a_1\right)^2 \, dx\, dy + \int_{inside(\phi)} \left(I(x, y) - a_2\right)^2 \, dx\, dy. \end{aligned} \tag{7.45}$$

Constants a_1 and a_2 represent the mean intensities of the interior and exterior of the segmented object(s). The energy $C(\phi, a_1, a_2)$ is minimized when the zero-level set ϕ coincides with the object boundary and best separates the object and background with respect to their mean intensities. Of course, other regional properties than image intensities can be used.

If the curve ϕ is outside the object, then $C_1(\phi) \approx 0$ and $C_2(\phi) \approx 0$. If the curve ϕ is inside the object, then $C_1(\phi) > 0$ and $C_2(\phi) > 0$. If the curve ϕ is both inside and outside the object, then $C_1(\phi) > 0$ and $C_2(\phi) > 0$ as shown in Figure 7.23.

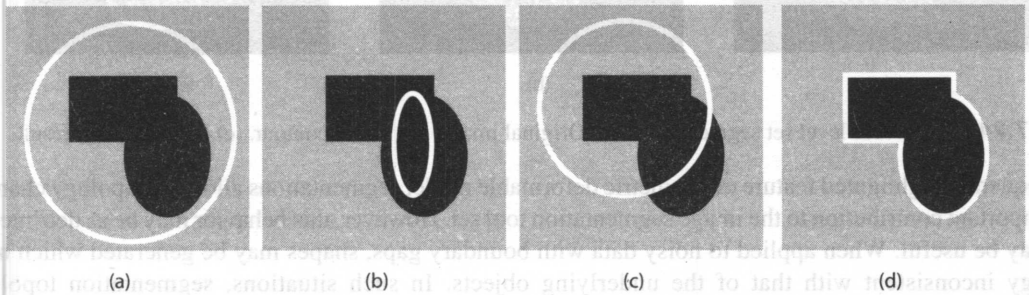


Figure 7.23: Chan-Vese energy functional. (a) $C_1(\phi) > 0$, $C_2(\phi) \approx 0$. (b) $C_1(\phi) \approx 0$, $C_2(\phi) > 0$. (c) $C_1(\phi) > 0$, $C_2(\phi) > 0$. (d) $C_1(\phi) \approx 0$, $C_2(\phi) \approx 0$.

In order to solve more complicated segmentation tasks, regularizing terms like length of the curve ϕ or the area of the region inside ϕ may be included, yielding an energy functional:

$$C'(\phi, a_1, a_2) = \mu \text{ (Length of } \phi) + \nu \text{ (Area inside } \phi)$$
$$+ \lambda_1 \int_{inside(\phi)} (I(x, y) - a_1)^2 \, dx \, dy + \lambda_2 \int_{outside(\phi)} (I(x, y) - a_2)^2 \, dx \, dy, \tag{7.46}$$

where $\mu \geq 0$, $\nu \geq 0$, $\lambda_1, \lambda_2 \geq 0$. The *inside* ϕ portion of image I corresponds to $\phi(x, y) > 0$ and *outside* ϕ corresponds to $\phi(x, y) < 0$. Using the Heaviside function $H(z)$

$$H(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}, \quad \delta_0 = \frac{dH(z)}{dz} \tag{7.47}$$

the level set equation minimizing the Chan-Vese energy functional C (equation 7.46) is

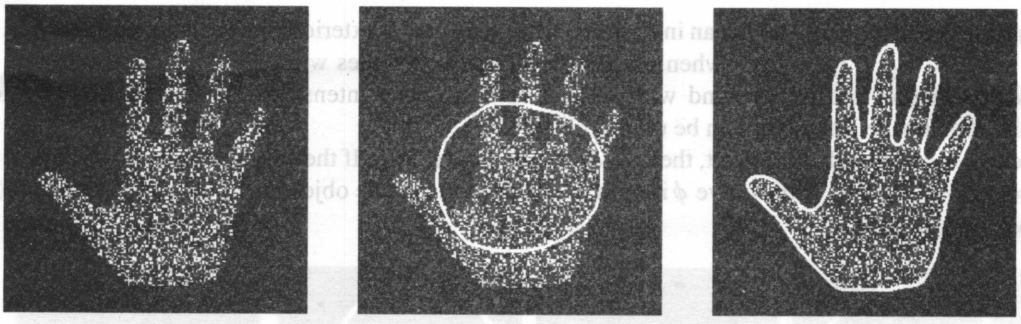
$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left(\mu \, \text{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (I(x, y) - a_1)^2 + \lambda_2 (I(x, y) - a_2)^2 \right). \tag{7.48}$$

The level set equation can be solved iteratively using time step Δt. However, inherent time step requirements exist to ensure stability of the numerical scheme via the Courant-Friedrichs-Lewy (CFL) condition [Heath, 2002]. In Chan and Vese's approach, the following time step can be used

$$\Delta t \leq \frac{\min(\Delta x, \Delta y, \Delta z)}{(|\mu| + |\nu| + |\lambda_0| + |\lambda_1|)}. \tag{7.49}$$

Figure 7.24 gives an example of a noisy image segmentation using the Chan-Vese energy equation (7.46). In this example, 2D curvature was taken as the approximation of div $(\nabla \phi / |\nabla \phi|)$

A large variety of applications exist in which geometric deformable models were used for image segmentation. Examples include a level set-based cortical unfolding method [Hermosillo et al., 1999]; cell segmentation [Sarti et al., 1996; Yang et al., 2005]; cardiac image analysis [Niessen et al., 1998; Angelini et al., 2004; Lin et al., 2003], and many others.



(a) (b) (c)

Figure 7.24: Chan-Vese level set segmentation. (a) Original image. (b) Initial contour. (c) Segmentation result.

The frequently highlighted feature of geometric deformable model segmentations allowing topology changes is an important contribution to the image segmentation tool set. However, this behavior may be as detrimental as it may be useful. When applied to noisy data with boundary gaps, shapes may be generated which have topology inconsistent with that of the underlying objects. In such situations, segmentation topology constraints may be required and a choice of parametric deformable models or graph-based approaches may be more suitable.

7.4 FUZZY CONNECTIVITY

Many image segmentation methods are based on crisp (or *hard-coded*) relationships between or within the individual regions to be segmented. In many cases, however, these relationships may vary across the image due to noise, uneven illumination, limited spatial resolution, partial occlusions, etc. **Fuzzy connectivity** segmentation approach takes these uncertainties into consideration. Rather than defining crisp relations, it attempts to describe the segmentation task with fuzzy rules such as *if two regions have about the same gray-value and if they are relatively close to each other in space, then they likely belong to the same object.* A framework for such a reasoning approach is called **fuzzy logic** and is treated in more detail in Section 9.1. While not essential for understanding the method described here, reading the fuzzy logic section may provide additional insight into this powerful concept.

Fuzzy connectivity segmentation is attempting to mimic the analysis strategy of a trained human observer who is typically able to perform a segmentation task by hand, frequently considering the likelihood of whether nearby image elements belong together. If they seem to belong to the same object based on their image and spatial properties, they are assigned to the same region. In other words, the image pixels seem to *hang together* when forming an object. The **hanging togetherness** property is then described using fuzzy logic. Early work in the fuzzy connectivity field was reported in [Rosenfeld, 1979, 1984], followed by [Bloch, 1993; Dellepiane and Fontana, 1995]. Udupa et al. stated an important concept that voxels belonging to the same objects tend to *hang together* thus defining objects by a combination of the spatial relationship of its elements (pixels, voxels), at the same time considering local image properties [Udupa and Samarasekera, 1996a; Udupa et al., 1997; Rice and Udupa, 2000; Saha et al., 2000]. The spatial relationships should be determined for *each* pair of image elements in the entire image. To accomplish that, local and global image properties are considered.

The *local* fuzzy relation is called **fuzzy affinity**, denoted by $\psi \in [0,1]$, and represents a strength of *hanging togetherness* of nearby image elements. Each such image element or **spatial element** is called a **spel**. (In 2D, a spel is equivalent to a pixel, in 3-D to a voxel.) Therefore, we will use *spel* and *image element* interchangeably. The *affinity* is a function of the spatial distance between two fuzzy adjacent image elements considering their image intensities or other image-derived properties (e.g., image edges). As such, any image I can be represented by a pair $I = (C, f)$, in which C represents the image domain and f represents local image properties. Then, $f(c) \in [0,1]$ represents a normalized image property (feature) associated with spel c. A more detailed treatment of fuzzy affinity is given below.

The **fuzzy adjacency** $\mu(c, d) \in [0,1]$ of two elements c, d is determined by the fuzzy adjacency function. *Hard*-adjacency results in binary adjacency values-spels that share a common face (e.g., 4-connectivity in 2D, 6-connectivity in 3D) are regarded as fully adjacent (*adjacency value* = 1). Any other spel pair is considered non-adjacent (*adjacency value* = 0). When using hard six-adjacency in 3D ($n = 3$) and considering two spels c and d, the binary adjacency values can be defined as

$$\mu(c,d) = \begin{cases} 1 & \text{if } c \text{ and } d \text{ are idntical } or \text{ of differ in exacly one coordinate by 1,} \\ 0 & \text{otherwise,} \end{cases} \tag{7.50}$$

A general n-dimensional fuzzy spel adjacency may be defined [Udupa and Samarasekera, 1996a]

$$\mu(c,d) = \begin{cases} \dfrac{1}{1 + k_1 \sqrt{\sum_{i=1}^{n} (c_i - d_i)^2}} & \text{if } \sum_{i=1}^{n} |c_i - d_i| \le n, \\ 0 & \text{otherwise,} \end{cases} \tag{7.51}$$

where k_1 is a nonnegative constant. Non-binary definitions of adjacency are possible with adjacency values ranging from 0 to 1. The affinity function $\psi(c, d)$ discussed above is only determined for spels c, d that are *fuzzy adjacent,* i.e. which have adjacency value $\mu(c, d) \neq 0$.

Fuzzy connectedness μ_ψ is a *global* fuzzy relationship that assigns every pair of image elements c and d a value in the interval [0,1] based on the affinity values ψ along all possible paths between these two image elements. The elements c and d are not expected to be nearby. They are connected by a path $\pi = (c^{(0)}, ..., c^{(N)})$ of spels, with $c = c^{(0)}$ and $d = c^{(N)}$. Each pair of consecutive spels is characterized by a fuzzy affinity $\psi\ (c^{(N)}, c^{(n+1)})$, $0 \le n < N - 1$. For each path, its strength is defined as the minimum affinity value of all pairwise consecutive elements on the path, so the strength of the entire path is defined by the strength of its weakest local connection, and is quantified by

$$\psi'(\pi) = \min_{0 \le n \le N-1} \psi\left(c^{(n)}, c^{(n+1)}\right). \tag{7.52}$$

Many different paths may connect two spels c and d. Let M denote the set of all path joining c, d. Note that M is not necessarily finite. The fuzzy connectedness is defined as i.e., the value of fuzzy connectedness

$$\mu_\psi(c, d) = \max_{\pi \in M} \psi'(\pi), \tag{7.53}$$

(global hanging togetherness) of c and d is determined as the maximum of the strengths of all possible paths between c and d. The strength of connectedness of all possible pairs of elements defining a fuzzy connected object is determined via dynamic programming [Udupa and Samarasekera, 1996a] (see Algorithm 7.5).

Starting from a seed-spel c and determining the fuzzy connectedness $\mu_\psi(c, d_i)$ to every other spel d_i in the image domain C, assigning the corresponding connectedness value to every spel, the resulting image is a fuzzy **connectedness map** representing the degree of connectedness of every spel in the image with the seed-spel c. Any degree of connectedness in the range [0,1] is possible. A very strong connectedness is denoted by 1, no connectedness by 0. By thresholding the connectedness map with an appropriate value, only spels with a certain pre-determined minimum degree of connectedness to the seed-spel remain. Thresholding the connectedness map yields the segmentation result.

Algorithm 7.4: Absolute fuzzy connectivity segmentation

1. Define properties of fuzzy adjacency and fuzzy affinity.
2. Determine the affinity values for all pairs of fuzzy adjacent spels.
3. Determine the segmentation seed element c.
4. Determine all possible paths between the seed c and all other image elements d_i in the image domain C (not forming loops) considering the fuzzy adjacency relationship.
5. For each path, determine its strength according as the minimum affinity along the path (equation 7.52).
6. For each image element d_j, determine its fuzzy connectedness $\mu_\psi(c, d_j)$ to the seed point c as the maximum strength of all possible paths $(c,..., d_j)$ (equation 7.53) and form an image connectedness map.
7. Threshold the connectedness map with an appropriate threshold t to segment the image into an object containing the seed c and the background.

The fuzzy affinity concept requires additional explanation. In most real-world applications, performance of fuzzy connectivity segmentation largely depends on the appropriate design of fuzzy affinity, which is computed using local image properties. Let the *fuzzy affinity* $\psi(c, d)$ quantify the hanging-togetherness of two spels c and d; by definition, $\psi(c, d)$ has to be reflexive and symmetric; transitivity is not required. The fuzzy affinity $y(c, d)$ is a function of fuzzy adjacency $\mu(c, d)$, spel properties $f(c)$, $f(d)$, and-in spatially variant cases-of c and d

$$\psi(c, d) = \frac{\mu(c, d)}{1 + k_2 \left| f(c) - f(d) \right|}, \tag{7.54}$$

where μ is the fuzzy adjacency defined by equation (7.51), k_2 is a nonnegative constant. A general affinity function can be defined as [Udupa and Samarasekera, 1996a]

$$\psi(c,d) = \begin{cases} \mu(c,d)\left(\omega h_1\left(f(c), f(d)\right) + (1-\omega)\, h_2\left(f(c), f(d)\right)\right) & c \neq d, \\ 1 & \text{otherwise,} \end{cases} \tag{7.55}$$

where ω is a weighting factor, h_1 and h_2 are segmentation task dependent and may be constructed from the following terms

$$g_1(f(c), f(d)) = \exp\left(-\frac{1}{2}\left(\frac{\frac{1}{2}[f(c) + f(d)] - m_1}{\sigma_1}\right)^2\right), \tag{7.56}$$

$$g_2(f(c), f(d)) = \exp\left(-\frac{1}{2}\left(\frac{|f(c) + f(d)| - m_2}{\sigma_2}\right)^2\right), \tag{7.57}$$

$$g_3(f(c), f(d)) = 1 - g_1(f(c), f(d)), \tag{7.58}$$

$$g_4(f(c), f(d)) = 1 - g_2 f(c), f(d)), \tag{7.59}$$

where m_1 and m_2 are mean values, and σ_1 and σ_2 standard deviations reflecting properties of the object of interest. These m and σ values can be calculated from the spels that are a priori known to belong to the object or background. Such a set of spels can either be provided by the user, or can be determined automatically using a rough pre-segmentation. As noted by [Udupa and Samarasekera, 1996a], $g_1(\cdot)$ and $g_2(\cdot)$ in equations (7.56) and (7.57) can also be expressed in a multivariate version.

Affinity function behavior can be influenced by the choice of the functions h_1 and h_2. For example, choosing $h_1(f(c), f(d)) = g_1(f(c), f(d))$, $\omega = 1$ favors spels that are closer to an expected mean value μ_1. Choosing $h_1(f(c), f(d)) = g_1(f(c), f(d))$, $h_2(f(c), f(d)) = g_4(f(c), f(d))$, and $\omega = 0.5$ decreases the affinity $\psi(c, d)$ if the gradient between the spels is close to the mean value μ_2. A slightly different affinity function can be found in [Carvalho et al., 1999]. An affinity function in which *homogeneity-based* and *object-feature-based* components are treated separately was proposed in [Saha and Udupa, 1999].

Finding the fuzzy connectedness $\mu_\psi(c, d)$ for every spel of the image domain $d \in C$, $c \neq d$ and assigning the respective connectedness value to every spel results in the *connectedness map* as introduced above. The following algorithm generates a connectedness map that can subsequently be thresholded at any value (possibly in an interactive way) as described in Algorithm 7.4. The algorithms 7.5 and 7.6 are based on dynamic programming [Udupa and Samarasekera, 1996a]. Generally, the algorithm outputs an image with values f_c expressing strength of connectivity between the seed-spell c and all other image spels $d \in C$.

Algorithm 7.5: Fuzzy object extraction

1. Define a seed-point c in the input image.
2. Form a temporary queue Q and a real-valued array f_c with one element $f_c(d)$ for each spel d.
3. For all spels $d \in C$, initialize array $f_c(d) := 0$ if $d \neq c$; $f_c(d) := 1$ if $d = c$.
4. For all spels $d \in C$ for which fuzzy spel adjacency $\mu_\psi(c, d) > 0$, add spel d to queue Q.
5. While the queue Q is not empty, remove spel d from queue Q and perform the following operations:
   $f_{max} := \max_{e \in C} \min(f_c(e), \psi(d, e))$
   if $f_{max} > f_c(d)$ then
   $f_c(d) : f_{max}$
   for all spels g for which $\psi(d, g) > 0$, add g to queue Q
   endif
   endwhile
6. Once the queue Q is empty, the connectedness map (C, f_c) is obtained.

Proof of convergence can be found in [Udupa and Samarasekera, 1996a]. As described in Algorithm 7.4, the connectedness map must be thresholded so that the segmented object only contains spels with an above-threshold connectedness to the seed-point. It can be shown that the resulting object is contiguous.

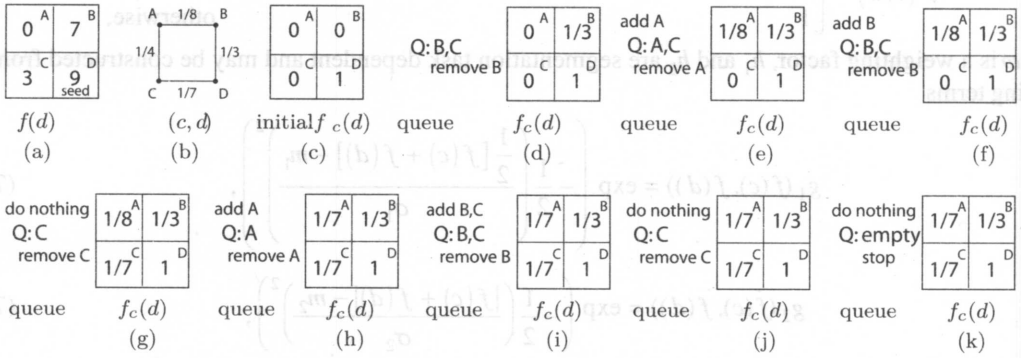


Figure 7.25: Fuzzy object extraction using Algorithm 7.5. (a) Image properties, which can be represented as image intensities. (b) Fuzzy affinity ψ (c, d) calculated according to equation (7.54), $k_2 = 1$. (c) Initialized array f_c (d). (d) Initial queue Q, temporary values of $f_c(d)$ after removal of spel B from queue Q. (e–j) Intermediate algorithm steps. (k) The queue Q is empty, stop. Values of array $f_c(d)$ represent the connectedness map.

A simple segmentation example associated with Algorithm 7.5 is given in Figure 7.25. Starting with a tiny 2×2 image the intensities of which are shown in Figure 7.25a and using the intensity as image features used for calculating the fuzzy affinity according to equation (7.54), Figure 7.25b gives the affinity map $\psi(c, d)$. Let the image spels (pixels) be identified with capital characters A,B,C,D–and let the seed c be located in pixel D. The initialization of array f_c can be seen in panel (c)-step (3) of Algorithm 7.5. Panel (d) shows the initial queue content according to step 4 of Algorithm 7.5. When B is removed from the queue $(d = B)$, $f_{\max}$ is calculated as given in step 5 examining all spels $e \in \{A, C, D\}$. For $e = A, C, D$

$$\begin{aligned} \min [f_c(A),\ \psi(B, A)] &= \min [0,1/8] = 0, \\ \min f_c(C),\ \psi(B, C)] &= \min [0, 0] = 0, \\ \min [f_c(D),\ \psi(B, D)] &= \min [1, 1/3] = 1/3, \end{aligned} \tag{7.60}$$

and the maximum $f_{\max} = 1/3$. Since $f_{\max} > 0$ ($> f_c(B)$), $f_c(B)$ is updated to 1/3 as shown in panel (d). As further specified in step 5 of the algorithm, $g = A$ is the only spel with non-zero affinity $\psi(d, g) = \psi(B, A)$ and is consequently added to the queue. Following the same set of operations of step 5, the intermediate states of the queue Q and array f_c are given in panels (e–j) of Figure 7.25. Note that some spels are repeatedly added to and removed from the queue. Once the queue becomes empty, the values held in array f_c represent the connectedness map that can be thresholded to obtain the segmentation result as given in step 7 of Algorithm 7.4.

If a lower bound of the connectedness map threshold is known beforehand, the slightly more efficient Algorithm 7.6 can be used. The increase in efficiency is given by value oft: the closer t is to 1, the higher the efficiency gain. Θ_t is used to denote a subinterval of [0,1] defined as

$$\Theta_t = [t, 1], \qquad \text{with } 0 \le t \le 1\ . \tag{7.61}$$

Algorithm 7.6: Fuzzy object extraction with preset connectedness

1. Define a seed-point c in the input image.
2. Form a temporary queue Q and a real-valued array f_c with one element $f_c(d)$ for each spel d.
3. For all spels $d \in C$, initialize array $f_c(d)$: = 0 if d = c; $f_c(d)$:= 1 if $d = c$.
4. For all spels $d \in C$ for which fuzzy spel adjacency $\mu_\psi(c, d) > t$, add spel d to queue Q.

5. While the queue Q is not empty, remove spel d from queue Q and perform the following operations:
   $f_{max} := \max_{e \in C} \min(f_c(e), \psi(d, e))$
   if $f_{max} > f_c(d)$ then
   $fc(d) := f_{max}$
   for all spels g for which $\psi(d, g) > 0$, add g to queue Q
   endif
   endwhile
6. Once the queue Q is empty, the connectedness map (C, f_c) is obtained.

Note that in both Algorithms 7.5 and 7.6, a spel may be queued more than once. This leads to the repeated exploration of the same subpaths and suboptimal processing time. A connectedness map generation approach based on Dijkstra's algorithm was reported in [Carvalho et al., 1999] in which Algorithm 7.5 was modified so that each spel gets queued at most once, and reported a 6- to 8-fold speedup.

The absolute fuzzy connectivity method suffers from problems similar to traditional region growing algorithms [Jones and Metaxas, 1997] and determining the optimal threshold of the connectivity map is difficult to automate. The absolute fuzzy connectivity method is however a foundation for the more powerful extensions to the basic method.

Relative fuzzy connectivity was introduced in [Saha and Udupa, 2000c; Udupa et al., 1999]. The main contribution of this approach is the elimination of the connectedness map thresholding step. Instead of extracting a single object at a time as described above, two objects are extracted by the relative fuzzy connectivity method. During the segmentation, these two objects are competing against each other with each individual spel assigned to the object with a stronger affinity to this spel.

The 2-object relative fuzzy connectivity method was later refined to include **multiple objects** in [Herman and Carvalho, 2001; Saha and Udupa, 2001; Udupa and Saha, 2001]. In [Saha and Udupa, 2001] the authors prove that simply using different affinities for different objects is not possible since this would mean that fundamental properties of fuzzy connectivity are no longer guaranteed. Instead, the affinities of the different

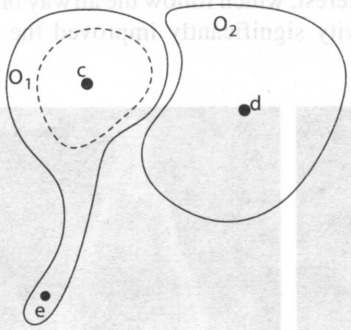


Figure 7.26: Segmentation task that can be solved by iterative fuzzy connectivity.

objects have to be combined into a single affinity. This is done by calculating the fuzzy union of the individual affinities. The extension to multiple object segmentation is a significant improvement compared to relative fuzzy connectivity.

Figure 7.26 demonstrates a situation in which fuzzy connectivity will probably fail to identify objects correctly. Two objects O_1 and O_2 are located very close to each other. Due to limited resolution, the border between O_1 and O_2 may be weak causing $\mu_\psi(d, e)$ to be of similar magnitude to $\mu_\psi(c, e)$. Objects O_1 and O_2 may thus be segmented as a single object. This problem can be overcome by considering **iterative fuzzy connectivity** [Saha and Udupa, 2000b; Udupa et al., 1999]. As can be seen from the figure, the optimal path between d and e probably passes through the core of O_1, depicted by a dashed line around c (Figure 7.26). This core can be segmented first, for example with the relative fuzzy connectivity algorithm. After that, paths for the object O_2 between two spels not located in this core (like d and e) are not allowed to pass through the core of O_1. The objects are segmented in an iterative process. In this approach, the same affinity function must be used for all objects.

Scale-based fuzzy connectivity approach considers neighborhood properties of individual spels when calculating the fuzzy affinity functions $\psi(c, d)$ [Saha and Udupa, 1999]. Calculating $\psi(c, d)$ is performed in two hyperballs centered at c and d, respectively. The scale of the calculation is defined by the radii of the hyperballs, which are derived from the image data based on the image content. The scale is thus adaptively varying and is location specific. This approach generally leads to an improved segmentation, however with a considerable increase in computational cost.

Fuzzy connectivity segmentation has been utilized in a variety of applications including interactive detection of multiple sclerosis lesions in 3D magnetic resonance images in which an improved segmentation reproducibility was achieved compared to manual segmentation [Udupa and Samarasekera, 1996a]. An approach for abdomen and lower extremities arterial and venous tree segmentation and artery-vein separation was reported in [Lei et al., 1999, 2000]. First, an entire vessel tree is segmented from the magnetic resonance angiography data using absolute fuzzy connectedness. Next, arteries and veins are separated using iterative relative fuzzy connectedness. For the artery-vein separation step, seed image elements are interactively determined inside an artery and inside a vein; large-aspect arteries and veins are separated, smaller-aspect separation is performed in an iterative process, 4 iterations being typically sufficient. To separate the arteries and veins, a distance transform image is formed from the binary image of the entire vessel structure (Figure 7.27a). Separate centerlines of arterial and venous segments between two bifurcations are determined using a cost function reflecting the distance transform values. All image elements belonging to the arterial or venous centerlines are then considered new seed elements for the fuzzy connectivity criterion thus allowing artery-vein separation. Figures 7.27b,c show the functionality of the method. A multi-seeded fuzzy connectivity segmentation based on [Udupa and Samarasekera, 1996b; Herman and Carvalho, 2001] was developed for robust detection of pulmonary airway trees from standard- and low-dose computed tomography images [Tschirren et al., 2005]. The airway segmentation algorithm presented here is based on *fuzzy connectivity* as proposed by Udupa et al. [Udupa and Samarasekera, 1996b] and Herman et al. [Herman and Carvalho, 2001]. During the execution of this algorithm, two regions-foreground and background-are competing against each other. This method has the great advantage that it can overcome image gradients and noise. The disadvantage is its relatively high computational complexity. Computing time can be reduced by splitting the segmentation space into small adaptive regions of interest, which follow the airway branches as they are segmented. The use of multi-seeded fuzzy connectivity significantly improved the method's performance in noisy low-dose image data, see Figure 7.28.

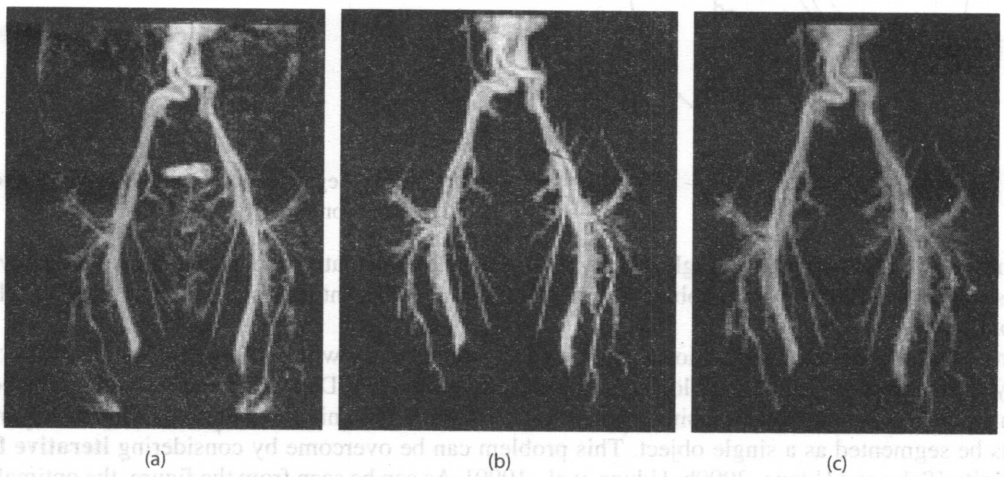
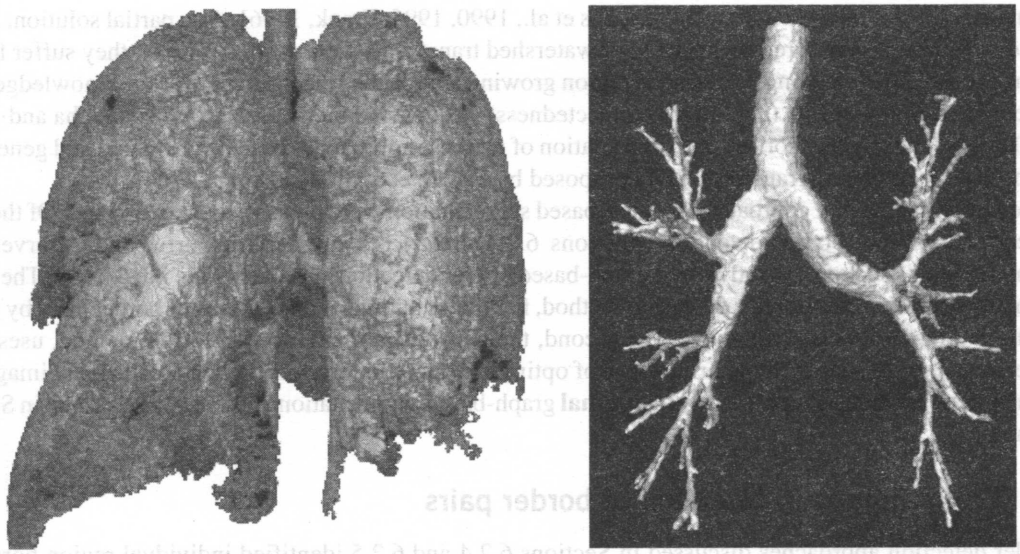


Figure 7.27: Segmentation and separation of vascular trees using fuzzy connectivity segmentation. (a) Maximum intensity projection image of the original magnetic resonance angiography data used for artery-vein segmentation in lower extremities. (b) Segmentation of the entire vessel tree using absolute fuzzy connectivity. (c) Artery–vein separation using relative fuzzy connectivity. *Courtesy of J. K. Udupa, University of Pennsylvania. A color version of this figure may be seen in the color inset—Plate 11.*



(a) (b)

Figure 7.28: Segmentation result using multi-seeded fuzzy connectivity approach. (a) Region growing segmentation results in a severe segmentation leak. (Emphysema patient, segmented with standard 3D region growing algorithm–the leak was unavoidable). (b) Multi-seeded fuzzy connectivity succeeded with the image segmentation using a standard setting of the method.

7.5 TOWARDS 3D GRAPH-BASED IMAGE SEGMENTATION

Graph-based approaches play an important role in image segmentation. The general theme of these approaches is the formation of a weighted graph $G = (V, E)$ with *node set* V and *arc set* E. The graph nodes are also called **vertices** and the arcs are called **edges**. The nodes $v \in V$ correspond to image pixels (or voxels), and arcs $(v_i, v_j) \in E$ connect the nodes v_i, v_j according to some neighborhood system. Every node v and/or arc $(v_i, v_j) \in E$ has a cost representing some measure of preference that the corresponding pixels belong to the object of interest.

Depending on the specific application and the graph algorithm being used, the constructed graph can be *directed* or *undirected*. In a directed graph (or *digraph*), the arcs (v_i, v_j) and (v_j, v_i) $(i \neq j)$ are considered distinct, and they may have different costs. If a directed arc (v_i, v_j) exists, the node v_j is called a *successor* of v_i. A sequence of consecutive directed arcs (v_o, v_1), (v_1, v_2),..., $\{v_{k-i}, v_k)$ form a directed path (or *dipath*) from v_0 to v_k.

Typical graph algorithms that were exploited for image segmentation include minimum spanning trees [Zahn, 1971; Xu et al., 1996; Felzenszwalb and Huttenlocher, 2004], shortest paths [Udupa and Samarasekera, 1996a; Falcao et al., 2000; Falcao and Udupa, 2000; Falcao et al., 2004], and graph-cuts [Wu and Leahy, 1993; Jermyn and Ishikawa, 2001; Shi and Malik, 2000; Boykov and Jolly, 2000, 2001; Wang and Siskind, 2003; Boykov and Kolmogorov, 2004; Li et al., 2004c]. Graph-cuts are relatively new and arguably the most powerful among all graph-based mechanisms for image segmentation. They provide a clear and flexible global optimization tool with significant computational efficiency. An approach to single and multiple surface segmentation using graph transforms and graph cuts was reported in [Wu and Chen, 2002; Li et al., 2006].

The introduction of graph-cuts into image analysis happened only recently [Boykov and Jolly, 2000; Kim and Zabih, 2003; Li et al., 2004b]. Classic optimal boundary-based techniques (e.g., dynamic programming, $A*$ graph search, etc.) were used on 2D problems. Their 3-D generalization, though highly desirable, has

been unsuccessful for over a decade [Thedens et al., 1990, 1995; Frank, 1996]. As a partial solution, region-based techniques such as region growing or watershed transforms were used. However, they suffer from an inherent problem of 'leaking.' Advanced region growing approaches incorporated various knowledge-based or heuristic improvements (e.g., fuzzy connectedness) [Udupa and Samarasekera, 1996a; Saha and Udupa, 2000a]. The underlying shortest-path formulation of all these approaches has been revealed and generalized by the Image Foresting Transform (IFT) proposed by Falcão et al. [Falcao et al., 2004].

Several fundamental approaches to edge-based segmentation were presented in Section 6.2. Of them, the concept of optimal border detection (Sections 6.2.4, 6.2.5) is extremely powerful and deserves more attention. In this section, two advanced graph-based border detection approaches are introduced. The first of them, the **simultaneous border detection** method, facilitates optimal identification of border pairs by finding a path in a three-dimensional graph. The second, the **sub-optimal surface detection** method, uses multi-dimensional graph search for determination of optimal surfaces in three- or higher-dimensional image data. Both of these methods paved the way to **optimal** graph-based segmentation approaches described in Sections 7.6 and 7.7.

7.5.1 Simultaneous detection of border pairs

Border detection approaches discussed in Sections 6.2.4 and 6.2.5 identified individual region borders. If the goal is to determine borders of elongated objects, it may be advantageous to search for the pair of left and right borders **simultaneously** [Sonka et al., 1993, 1995]. Such an approach facilitates more robust performance if the borders forming the border pair are interrelated, allowing information about one border to help identify the second. Examples include situations in which one border is locally noisy, ambiguous, or uncertain, where identifying borders individually may fail. Following a border of a road or river in a satellite image is an example. As seen in Figure 7.29a, the left and right borders, if considered individually, seem to be reasonable. However, if taken as a pair, it is unlikely that they represent left and right borders of, say, a river. Obviously, there is information contained in the position of one border that might be useful in identifying the position of the other border, and more probable borders may be detected if this is considered (Figure 7.29b).

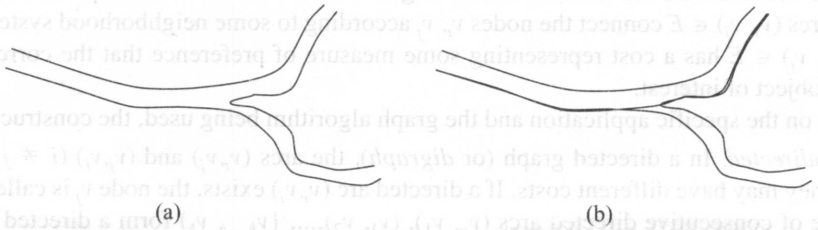


Figure 7.29: Individual and simultaneous border detection. (a) Individually identified borders may not be reasonable as a pair. (b) Simultaneously identified borders satisfy border-pair properties.

To search for an optimal border pair, the graph must be three-dimensional. Shown in Figure 7.30a are two adjacent but independent two-dimensional graphs, nodes in which correspond to pixels in the straightened edge image (Section 6.2.4). The column of nodes separating the left graph and the right graph corresponds to the pixels on the approximate region centerline. A row of nodes in the left graph corresponds to the resampled pixels along a line perpendicular to and left of the region centerline. If we connect nodes in the left graph as shown in Figure 7.30a, the resulting path corresponds to a possible position for the left border of the elongated region. Similarly, linking nodes together in the right graph produces a path corresponding to a possible position of the right region border. If the conventional border detection methods that were described earlier are applied, the 2D graphs would be searched independently to identify optimal left and right region borders.