A New Image Segmentation and Smoothing Method Based on the Mumford–Shah Variational Model

Song Gao

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Computer Science

Concordia University

Montreal, Quebec, Canada

December 2003

Your file Votre référence
ISBN: 0-612-91033-4
Our file Notre référence
ISBN: 0-612-91033-4

ABSTRACT

A New Image Segmentation and Smoothing Method Based on the Mumford–Shah Variational Model

Song Gao

Recently Chan and Vese have developed an active contour model for image segmentation and smoothing. Tsai et al. have also developed a similar approach independently. In this thesis, we develop a new hierarchical method which has many advantages compared to the Chan and Vese multiphase active contour models. *First*, unlike previous works, the curve evolution partial differential equations (PDEs) for different level set functions are decoupled. Each curve evolution PDE is the equation of motion of just one level set function; and different level set equations of motion are solved in a hierarchy. This decoupling of the motion equations of the level set functions speeds up the segmentation process significantly. *Secondly*, because of the coupling of the curve evolution equations associated with different level set functions, the initialization of the level sets in Chan and Vese's method is difficult to handle. The hierarchical method proposed in this thesis can avoid the problem due to the choice of initial conditions. *Thirdly*, we use the diffusion equation for denoising. This method therefore can deal with very noisy images. In general, our method is fast, flexible, not sensitive to the choice of initial conditions, and produces very good results.

Acknowledgements

I would like to gratefully acknowledge my supervisor Dr. T. D. Bui for introducing me to the fascinating fields of the level set, variational principles, and partial differential equations, diffusion methods for image processing and support during this work. His enthusiastic guidance of my work and carefully read all words of this thesis were truly invaluable.

I would like also to thank Dr. C. Y. Suen and Dr. S. P. Mudur for their excellent classes in Pattern Recognition and Image processing, and Advanced Computer Graphics. I learned a lot from them.

Finally, I am forever indebted to my wife, Xiuli Ge for her understanding, endless patience and encouragement when it was most required. I am also grateful to my daughter, Shiwei for her understanding.

Table of Contents

List of Figures

Mathematical Notation and Symbols

Ω	An open subset of R^N
$K^{0,\gamma}(\Omega)$	For $0 < \gamma \leq 1$: space of continuous functions f on Ω such that $\|f(x) - f(y)\| \leq c\,\|x - y\|^{\gamma}$, for some constant c, x, $y \in \Omega$. It is called the space of Hölder continuous functions with exponent γ.
$K^{n,\gamma}(\Omega)$	Space of n–times continuously differentiable functions whose nth partial derivatives belong to $K^{0,\gamma}(\Omega)$.
∇f	Gradient of f in the classical sense.
$\nabla \cdot \vec{A}$	Divergence operator: $\nabla \cdot \vec{A} = \sum_{i=1}^{N} \frac{\partial A_i}{\partial x_i}$.
$\nabla^2 f$	Laplacian operator: $\nabla^2 f = \sum_{i=1}^{N} \frac{\partial^2 f_i}{\partial x_i^2}$.
χ_R	Characteristic function of R: $\chi_R(x) = \begin{cases} 1 & \text{if } x \in R, \\ 0 & \text{otherwise.} \end{cases}$

Chapter 1

Introduction

IMAGE segmentation [1] and smoothing are two popular problems in image processing and computer vision [2]. The Mumford and Shah variational model [3] is one of the most widely studied mathematical models that can achieve both goals simultaneously by using a piecewise smooth representation of an image [4], [5]. Mumford–Shah variational methods have been extensively used in image processing because of their flexibility in modeling the image related problems, and various advantages in numerical implementations. The basic idea of variational methods is to minimize an energy functional $F(u(x,y), C(x, y))$ that contains boundary $C(x, y)$ and intensity value $u(x, y)$ of an image. If $C(x, y)$ is a closed curve, the image domain Ω is partitioned into sub-domains R and $\overline{R}$. The classical way to solve the minimization problem is to solve the corresponding Euler–Lagrange equation, which is a second order *partial differential equation* (PDE). Variational methods have been used in many areas such as image segmentation, object tracking, texture synthesis and vector field visualization [6].

The image segmentation problem under the framework of the Mumford and Shah variational functional [3] consists in computing a decomposition of the image domain $\Omega = \Omega_1 \cup \ldots \cup \Omega_n$ such that the image $u(x, y)$ varies smoothly and/or slowly within each region Ω_i $(i = 1, \ldots, n)$ but varies discontinuously and/or rapidly across the boundary C

between two adjacent regions. From the viewpoint of approximation theory, the segmentation problem can be stated as seeking ways to define and compute optimal approximations of a general image function $u(x, y)$ by piecewise smooth functions $u_i(x, y)$ in different region Ω_i.

Over the past decade, many different image segmentation approaches have been developed, such as the classical snakes model [7], geodesic active contours [8], curve evolution based on the Mumford–Shah functional method [5], and active contours without edges [4], [9]–[14]. A problem related to image segmentation is the object detection problem by snakes or active contours. An initial curve evolves in an image and stops on the boundaries of objects within the image. The classical active contour models [7], [8] use the gradient of the image for stopping criteria. The active contours (or snakes) stop on the boundaries where the magnitude of the gradient of the image is large. These active contour models are usually called boundary–based models.

For image smoothing, the technique developed from partial differential equation has become a widespread field of research. Especially the anisotropic diffusion model originally introduced by Perona and Malik [15], and further developed by [16], [17]. Other anisotropic diffusion techniques such as curve evolution methods based on geometric scale space [18]–[20], and construction of the diffusion tensor that contains the information of both modulus and direction of the gradient depending upon application goals [21]–[25], have also been developed. The basic idea of most anisotropic diffusion techniques is to employ the gradient (both modulus and direction) of the intensity to detect the edges between regions then smooth the image within the homogeneous region and along the edges but not across the boundaries of such regions.

Recently, Chan and Vese developed an active contour model to deal with the problem of image segmentation and smoothing by using piecewise constant (and/or smooth) representation of an image [4], [9]. Tsai, Yezzi and Willsky [5] also independently developed a segmentation and smoothing method very similar to that of the Chan and Vese piecewise smooth approach. These models are based on the Mumford–Shah variational model and are applied to solve the image segmentation and smoothing problems that can be formulated by *level set methods* introduced by Osher and Sethian in [26], and further developed in [27]–[34]. These models actually are *active contours based on the Mumford-Shah model with level set methods*. Their energy functional $F(u(x, y), C(x, y))$ contains a closed segmenting curve C and image data inside (denoted by u^{I}) and outside (denoted by u^{II}) the curve. Minimizing the energy functional F with respect to u^{I}, u^{II} and C, we can obtain the curve evolution PDE, and the optimal estimation equations for u^{I} and u^{II}. The motion of the curve is obtained by solving the curve evolution PDE, and the curve stops on the edges of the objects within a given image. The optimal estimation equations for u^{I} and u^{II} have a smoothing effect on the original image u_0, therefore u^{I} and u^{II}, which are obtained by solving the corresponding optimal estimation PDEs for u^{I} and u^{II}, are piecewise smooth approximations of u_0 inside and outside curve C respectively. The curve evolution PDE together with the optimal estimation PDEs produce piecewise smoothing and segmentation of an image simultaneously [4], [5], but shortcomings also exist in this approach. Because of the coupling between the curve evolution PDE and the optimal estimation PDEs for u^{I} and u^{II}, when we solve the curve evolution PDE the solutions of u^{I} and u^{II} are needed. Therefore it involves solving the optimal estimation PDEs for u^{I} and u^{II} at each curve evolution step, and that makes the

algorithm very slow. Since the segmentation of an image finishes at the final iteration, only the solutions of u^{I} and u^{II} at the final iteration are required in the piecewise smooth reconstruction of the image. All of the solutions of u^{I} and u^{II} before the final curve evolution steps are only required for solving the curve evolution PDE. Moreover, the optimal estimation PDEs for u^{I} and u^{II} are Poisson equations, their capability of denoising is limited.

Apart from the computational costs and the denoising qualities, handling the initial conditions correctly is another problem of this model in multiphase approach. As mentioned in [4], in the implementation through level set method, using one level set function we can represent only two phases in an image. In order to represent images with more complicated features, multiple level set functions should be used. Because the Mumford–Shah problems are non–convex, and because there is no uniqueness for the minimizer, the final segmented results may depend on the choices of the initial curves. The multiple seed initialization used by Chan and Vese [4] cannot always give correct results.

In this thesis, we propose a new algorithm for image segmentation and smoothing. Based on the Chan–Vese piecewise constant segmentation model and the level set method, we propose a new hierarchical method of multiphase level set framework for piecewise constant segmentation of images. The multiphase segmentation of an image is divided into different stages; at each segmentation stage only one curve evolution equation (equivalent to one level set equation) is used. The next segmentation stage begins after the previous stage has been completed. This continues until the last stage. The number of stages is the same as the number of level set functions. This hierarchical

segmentation method makes the algorithm fast and initial conditions easy to handle. We use the diffusion equation for image denoising because it can deal with very noisy images. The proposed method actually works in two steps: for a given image, we first apply the hierarchical piecewise constant segmentation method to partition the image, then apply diffusion filtering to different regions independently, but not across the boundaries of such regions. This method is fast, more flexible, and not sensitive to the initial conditions.

Thus, the contributions in this thesis are the development of new, efficient techniques for image segmentation and smoothing. This method can segment and smooth an image simultaneously and relatively enhance edges within the image.

This thesis is organized as follows: In chapter 2, we cover most of the mathematics used in this thesis, including the principle of variation, partial differential equations, and the level set methods. We survey some of the most popular and traditional image segmentation methods as well as the segmentation algorithms based on the energy functional minimization, and the basic idea of the anisotropic diffusion techniques for image smoothing and denoising in chapter 3. In chapter 4, we introduce our hierarchical image segmentations by using the level set method based on the Chan–Vese piecewise constant active contours model [9]. We present the two–step segmentation and smoothing method for image segmentation, smoothing and edge enhancement, and discuss how the proposed method works for images with or without noise in chapter 5. Chapter 6 is our experimental results, and the final chapter is our conclusion. The numerical implementation methods used in this thesis are presented in the appendix.

Chapter 2

Mathematical Preliminaries and the Level Set Method

This chapter introduces the mathematics used in this thesis. It covers some of the main ideas of the calculus of variations and the theory of partial differential equations (PDEs). The level set method that is the numerical technique used in most of the implementations in the thesis is also reviewed in this chapter.

2.1 The Calculus of Variations

The calculus of variations originates from well-known facts, such as the shortest path between two fixed points is a straight line, and the circle is the figure with the largest enclosed area inside a given perimeter. The calculus of variations is concerned with the extrema of a given functional (function of functions). It seeks to find the path, curve, or surface, for which a given functional has a minimum or maximum. Mathematically, this involves finding stationary values of a form of the functional integrals such as,

$$S[y] = \int_{x_1}^{x_2} F(x, y, y')dx, \tag{2.1}$$

where $y = y(x)$, and $y' = dy/dx$. In order to minimize (or maximize) the value of a functional $S[y]$ over the interval $[x_1, x_2]$, we must find the stationary values such that the variation of $S[y]$ equals zero ($\delta S = 0$) for any small change in $y(x)$ which is,

$$\delta S[y] = \int_{x_1}^{x_2} \delta F(x, y, y')dx, \tag{2.2}$$

Calculate the variation of the functional F we have,

$$\delta F = \frac{\partial F}{\partial y}\delta y + \frac{\partial F}{\partial y'}\delta y' = \frac{\partial F}{\partial y}\delta y + \frac{\partial F}{\partial y'}\frac{d}{dx}(\delta y) = \frac{\partial F}{\partial y}\delta y - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right)\delta y + \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\delta y\right). \tag{2.3}$$

Where $\delta y' = \delta\left(\frac{dy}{dx}\right) = \frac{d}{dx}(\delta y)$, and after integrating (2.3) by parts, we have,

$$\delta S = \int_{x_1}^{x_2}\left[\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right)\right]\delta y dx + \left[\frac{\partial F}{\partial y'}\delta y\right]_{x_1}^{x_2} = 0. \tag{2.4}$$

The variation requires that $\delta y(x_1) = \delta y(x_2) = 0$, therefore, $\left[\frac{\partial F}{\partial y'}\delta y\right]_{x_1}^{x_2} = 0$.

So we have:

$$\delta S = \int_{x_1}^{x_2}\left[\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right)\right]\delta y dx = 0. \tag{2.5}$$

Since δy is arbitrary we have the following equation,

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right) = 0, \tag{2.6}$$

which is called *Euler–Lagrange* partial differential equation. S has an extremum only if the *Euler–Lagrange* equation is satisfied. Solving the appropriate Euler–Lagrange equation is equivalent to solving the problem in the calculus of variations [35], [36].

2.2 Partial Differential Equations

On the basis of their characteristic or curves of information propagation, partial differential equations are usually classified into three categories, hyperbolic, parabolic, and elliptic [37], [38]. In this thesis, we only focus on the parabolic type equations. A typical parabolic equation is the *diffusion* equation as follows,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D \frac{\partial u}{\partial x} \right), \tag{2.7}$$

where D is the diffusion coefficient. It defines an initial value or Cauchy problem: If information on u is given at some initial time t_0 for all spatial variable x, then the evolution of $u(x, t)$ with time t is governed by this equation. The initial condition is usually specified as,

$$u(x, t = 0) = u_0(x). \tag{2.8}$$

The PDEs that govern the curve evolution in the image segmentation and used for diffusion filtering in the smoothing approach are the type of parabolic equation. The diffusion equation may, with or without boundary conditions, depend on the applications. Generally, we need to solve this equation numerically, and we will discuss some finite difference numerical techniques in the appendix.

If D in (2.7) is a constant, it is easy to get the analytical solution of the diffusion equation with the initial condition (2.8). For example, we can use the *Fourier transform method* to solve this diffusion equation with ($-\infty<x<\infty$). Applying Fourier transformation with respect to x to (2.7) and (2.8), and letting

$$\tilde{u}(\omega, t) = F[u(x,t)] = \int_{-\infty}^{\infty} u(x,t) e^{i\omega x} dx, \quad \tilde{u}_0(\omega) = F[u_0(x)] = \int_{-\infty}^{\infty} u_0(x) e^{i\omega x} dx.$$

From the time-differentiation property of Fourier transform [If the function f(x) has a Fourier transform $\tilde{u}(\omega)$, then its derivative f'(x) has the Fourier transform $i\omega\tilde{u}(\omega)$.] we have,

$$\frac{d\tilde{u}}{dt} = -D\omega^2\tilde{u}\,, \tag{2.7}'$$

$$\tilde{u}(\omega,0) = \tilde{u}_0(\omega)\,. \tag{2.8}'$$

Equation (2.7)′ together with (2.8)′ is an *ordinary differential equation* (ODE), its solution is, $\tilde{u}(\omega,t) = \tilde{u}_0(\omega)e^{-D\omega^2 t}$, therefore the solution of (2.7) together with (2.8) can be obtained by the *inverse Fourier transformation*, $u(x,t) = F^{-1}[\tilde{u}(\omega,t)]$. Employing the *convolution property* of the Fourier transform and do some straightforward calculations, we have the final solution,

$$u(x,t) = \frac{1}{2\sqrt{\pi Dt}}\int_{-\infty}^{\infty} u_0(\xi)e^{-\frac{(\xi-x)^2}{4Dt}}\,. \tag{2.9}$$

The idea behind the use of the diffusion equation in image processing arose from the use of the Gaussian filter in image smoothing. Convolving an image u_0 with a Gaussian filter $K_\sigma(x, y)$ as,

$$K_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{|x|^2+|y|^2}{2\sigma^2}\right) \tag{2.10}$$

with the standard deviation σ, is equivalent to the solution of the diffusion equation in *two spatial dimension* at time $t = 0.5\,\sigma^2$.

2.3 The Level Set Methods

The level set methods are the numerical techniques originally introduced by Osher and Sethian for analyzing and computing the motion of interfaces [26]. Many applications of the level set and PDE methods for image processing have also been developed [27], [29]–[31]. The basic idea of the *Level Set Methods* is the following: A boundary curve moving in the plane (e .g. 2–dimension) is replaced by a problem in 3–dimension (see Fig 2.1) through the introduction of the *level set function* $\phi(x, y, t)$. The equation of motion for the level set function is governed by a *partial differential equation*. We give the definition of the level set function and its properties next, followed by the introduction of the equation of motion for the level set function.

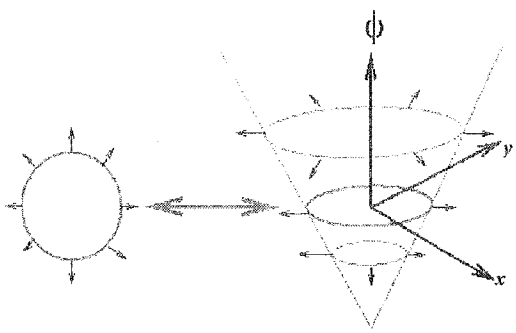


Fig. 2.1. Basis of the level set methods. A closed curve can be seen as the zero–level of a higher dimensional function ϕ [27].

2.3.1 The Level Set Function

From Fig. 2.1 we can see that an original circle in the 2–D xy plane can be represented by the *intersection* of a 3–D cone–shaped surface and a xy plane at zero height. More precisely, let us define a *level set function* $z = \phi(x, y, t)$ where the cross section in the xy plane and the level set function $\phi(x, y, t)$ represents the boundary $C(t)$ of the set where

$\phi(x, y, t) = 0$. Therefore, the level set function $\phi(x, y, t)$ has the following properties if the boundary curve $C(t)$ bounds a region Ω,

$$\begin{aligned} &\phi(x, y, t) > 0 && (x, y) \text{ inside } \Omega, \\ &\phi(x, y, t) < 0 && (x, y) \text{ outside } \Omega, \\ &\phi(x, y, t) = 0 && (x, y) \text{ on } \partial\Omega = C(t). \end{aligned} \tag{2.11}$$

For example, using an implicit function ϕ as follows can represent a curve in 2–dimensional space that is shown in Fig. 2.2.

$$\phi(x, y) = x^2 + y^2 - 1 \tag{2.12}$$

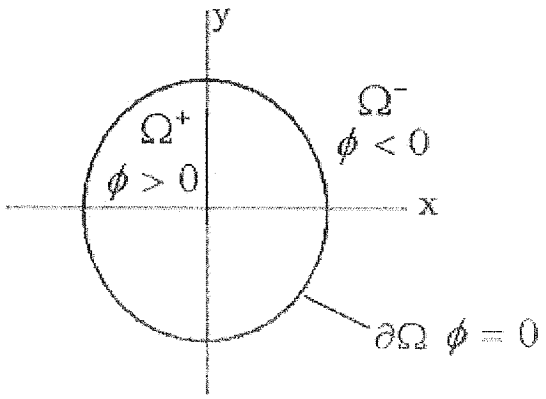


Fig. 2.2. Implicit representation of the circle $x^2 + y^2 = 1$.

In this two spatial dimensional case, the curve is called the lower–dimensional interface that separates R^2 into two (or more) sub-domains with nonzero areas. In Fig. 2.2 the interface curve is a *closed* curve, therefore it clearly has defined interior (e.g. $\phi > 0$) and exterior (e.g. $\phi < 0$) regions. To allow modeling of *dynamic* processes which means change of level set functions over time, a time–dependent PDE is introduced for the moving interface (*boundary curve*) [32]. We will discuss this equation of motion next.

2.3.2 The Level Set Motion Equation

We can link the motion of the boundary to the evolution of the level set function ϕ through an initial value problem where the boundary is given by the *zero level set* of the time–dependent level set function $\phi(x, y, t)$ at any time t. Thus, at any time t the level set value of each point $\vec{x}(t)$ on the boundary $\partial\Omega$ is given by $\phi(\vec{x}(t),t)$, must be zero (i.e. $\phi(\vec{x}(t),t)=0$) that implies $d\phi(\vec{x}(t),t)/dt=0$. By the chain rule we have,

$$\frac{\partial\phi(\vec{x},t)}{\partial t}+\nabla\phi\cdot\frac{d\,\vec{x}}{dt}=0. \tag{2.13}$$

It is an initial value partial differential equation and can be rewritten as,

$$\frac{\partial\phi(\vec{x},t)}{\partial t}+F\,|\nabla\phi|=0, \tag{2.14a}$$

where, $F=\vec{n}\cdot\vec{\mathrm{v}},(\vec{\mathrm{v}}=d\vec{x}/dt$ is the velocity of the boundary, $\vec{n}$ is the unit normal to the boundary) is the *normal velocity* of the boundary. We have to add the following conditions associated with (2.14a).

1. A boundary condition: we usually choose the normal derivative vanishes on the boundary $\partial\Omega$,

$$\frac{\partial\phi}{\partial\vec{n}}=0 \text{ on } \partial\Omega. \tag{2.14b}$$

2. An initial condition at $t = 0$. The initial function ϕ_0 usually associated with an initial given curve C_0:

$$\phi(x,y,0)=\phi_0(x,y). \tag{2.14c}$$

The equation (2.14a) is the *level set equation* introduced by Osher and Sethian [26]. This equation describes the evolution of the level set function ϕ in such a way that the

zero level set of this evolving function is always identified with the propagating boundary [27]. This means at any time t the moving boundary $C(t)$ is just the zero level set of the level set function $\phi(\vec{x}(t),t)$ which is the solution of the level set equation (2.14).

There are many advantages [27] to working with this perspective on propagating boundaries:

- The topological changes in the evolving boundary such as merging and breaking are handled naturally: The position of a boundary at time t is given by the zero level set of the evolving function ϕ, and it can merge as time advances (see Fig. 2.3).
- It is easy to build accurate numerical schemes to approximate the equations of motion (2.14). For example, the finite difference approximations can be used.
- Intrinsic geometric properties of the boundary such as the normal vector and the curvature can be easily expressed in terms of function ϕ. At any point of the boundary, the normal vector is given by

$$\vec{n} = \frac{\nabla\phi}{|\nabla\phi|}, \tag{2.15}$$

and the curvature at any point of the boundary is

$$\kappa = \nabla \cdot \vec{n} = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{\left(\phi_x^2 + \phi_y^2\right)^{3/2}}. \tag{2.16}$$

- The above 2–dimensional formulation can be extended to the applications in 3 or higher dimension.

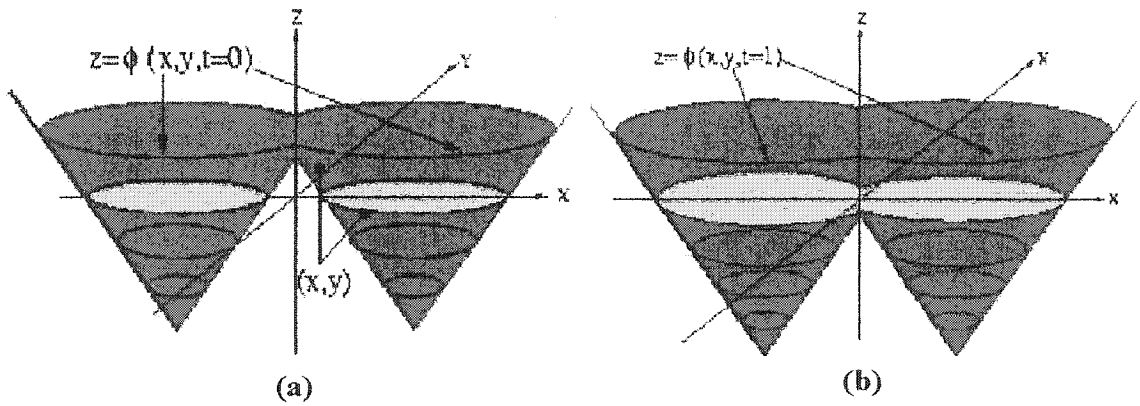


Fig. 2.3. The change of topology. (a) The *zero* level set at time $t = 0$ is the cross section of the level set surface and the *xy* plane. (b) Later in time ($t = 1$) the level set surface has moved and the new *zero* level set defines the new contour [27].

2.3.3 Curve Evolution with the Level Set Method

We illustrate the curve evolution governed by the solutions of the level set PDE through two examples with different propagating speed F of the interface, namely motion under curvature [26], [39] and motion with constant speed.

a) Motion under Mean Curvature

If the velocity in (2.14a) is chosen as $F = -\kappa$, the level set formulation of the curve evolution equation becomes:

$$\frac{\partial \phi(\vec{x},t)}{\partial t} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi|, \tag{2.17}$$

which is a non-linear diffusion equation. If $|\nabla\phi|$ = *constant*, (2.17) becomes linear (Gaussian) diffusion equation.

In the first example we consider the curve motion under the speed of mean curvature. Using the numerical methods in the appendix to solve the level set PDE (2.17) we obtain the moving curve at different time stages which are shown in Fig 2.4. (a) is the initial

curve, (b) to (e) are the resulting curves at different time stages. Notice that if we let the evolutions run until convergence, any curve changes to a circle and then collapses.

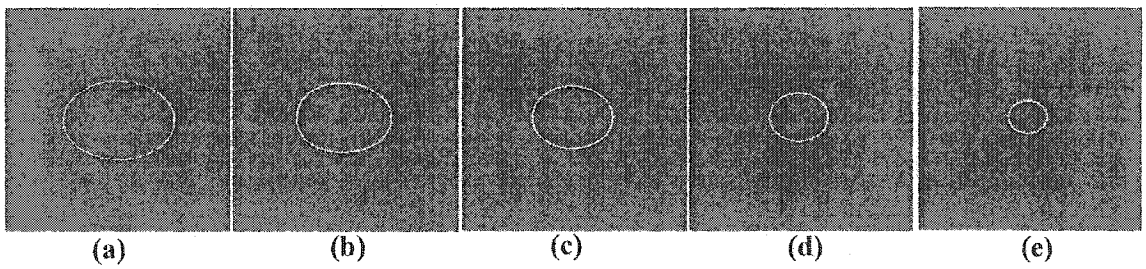


Fig. 2.4. Motion of a curve with a curvature–dependent speed. (a) Initial elliptic curve. (b)–(e) The moving curves at time t =1, 2, 3, 4 respectively.

These experimental results support the following famous theorem in differential geometry [26].

Theorem: *Any simple closed curve moving under its curvature collapses nicely to a circle and then disappears.*

b) Constant Speed Evolution [27], [32]

The second example is given by

$$\frac{\partial \phi}{\partial t} = c\,|\nabla \phi\,|, \tag{2.18a}$$

$$\phi(x, y, 0) = \phi_0(x.y)\,, \tag{2.18b}$$

where c is a constant. This equation represents a motion in the direction normal to the interface (boundary). For c = 1, the interface moves in the normal direction, it is the simulation of the wave–interface propagation of a *grass fire*. Fig. 2.5 shows the evolution of an elliptic curve under the speed c = 1, it moves normal to itself in the outward direction. The initial curve is shown in Fig. 2.5 (a), the other figures are the evolution of the curve at different time steps.

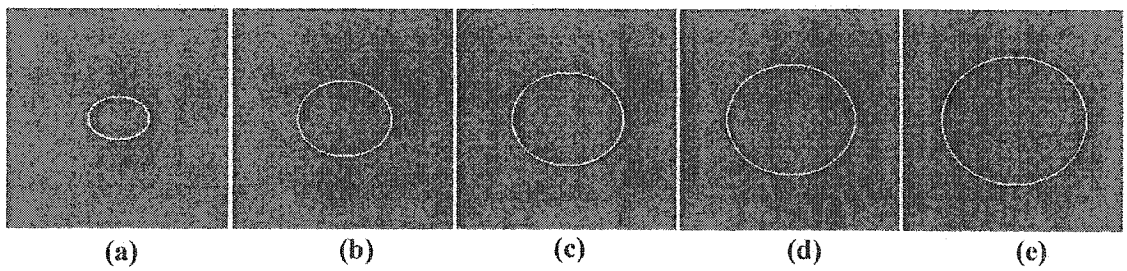


Fig. 2.5. Motion of a curve with speed c = 1. (a) Initial elliptic curve. (b)–(e) The moving curves at time t =4, 6, 8, 10 respectively.

If the speed $c = -1$, the interface moves opposite to the normal direction. In Fig. 2.6 we show the curve evolving with time in this case. Since the curve moves in the direction to the interior region defined by this curve, it shrinks with time but has different behavior compared to the mean curvature motion as shown in Fig. 2.4.

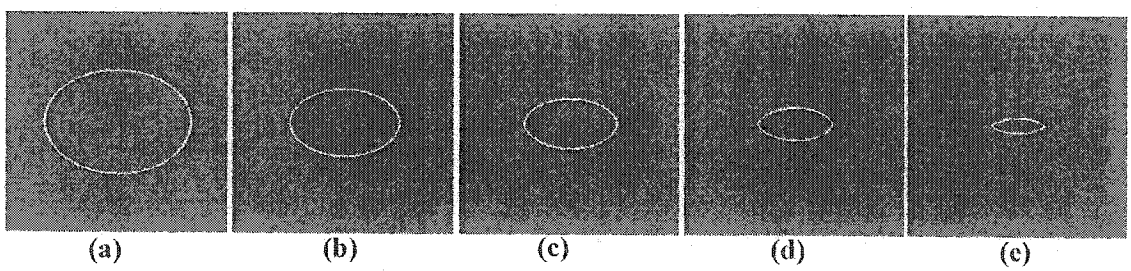


Fig. 2.6. Motion of a curve with speed c = −1. (a) Initial elliptic curve. (b)–(e) The moving curves at time t =4, 6, 8, 10 respectively.

The basic level set method concerns the solution of the level set PDE throughout the space. Clearly this is wasteful for the boundary evolution problem which only cares about the region near the zero level set. A local level set approach called *narrow banding* was developed to improve the computation costs [27], [29]. The narrow band approach defines ϕ only near the zero level set correspoding to a thin band around the boundary. We may solve the level set equation in a neighbourhood of C with a width $n\Delta x$, where n

is typically equal to 5 or 6. Points outside of this neighborhood need not be updated by this motion. The narrow band approach is a fast numerical method for the boundary evolution. It also allows the 3–dimensional interface evolution problem to be handled with ease.

Chapter 3

Image Segmentation and Smoothing Problems

In this chapter, we survey some of the classical image segmentation and smoothing techniques. We examine some of the widely used traditional image segmentation methods and some more recent variational functional based segmentation techniques, followed by the discussions of the PDE–based diffusion methods for image smoothing.

3.1 Image Segmentation Methods

3.1.1 Approaches to Image Segmentation

Image segmentation is one of the most important and difficult tasks in image processing. The aim is to subdivide an image into its constituent regions, and the subdivision should stop when the regions of interest have been detected. Traditional image segmentation algorithms are based on one of two basic properties of image intensity values which are similarity and discontinuity. The principal segmentation approaches in the first case are based on the portioning of an image into regions that are similar according to a set of predefined criteria. Thresholding, region growing, and region splitting and merging are examples of methods in this category. In the second case, segmenting an image is based

on the abrupt changes in the intensity values of the images, such as edges in an image. Classical d*iscontinuity detection* techniques, snakes and geometric active contour methods are examples in that category.

Discontinuity detections, such as points, lines, and edges detections are important in discussion of image segmentation. Since edges are the most important features in a digital image, *edge detection* in particular has been a main concern of segmentation algorithms for many years. Classical edges detection approaches are based on the first and second derivatives of the image. For the first–order derivatives, the earliest methods were based on the convolution of a mask with an image. Numerous masks have been developed over the past decades, such as Roberts gradient operators [40], Prewitt operators [41], Sobel operators [42], and Canny algorithm [43]. Further developments along this direction have been done by Deriche [44], and Shen and Castan [45]. Review materials on this topic can be found in [2], [46]. For second–order derivatives, the most widely used edge detection algorithm was proposed by Marr and Hildreth [47] based on zero–crossing detection of the *Laplacian of a Gaussian* (*LoG*) [48]–[50]. Corners are rounded and the connectivity at the junctions is poor in this kind of algorithm. These edge detection approaches are based on the local gradient or Laplacian of an image, and identify the edges that are characterized by the sharp changes of the image intensity.

Thresholding techniques [2], [51] are the early and popular approaches for image segmentation, because they are simple to implement. Thresholding techniques require very little computation, but the segmentations are often poor when there is noise in an image. Noise creates artificial boundaries where there were no anatomical boundaries in the original image, and the weak but real anatomical boundaries are often lost

completely. Further, due to noise boundaries may contain artifactual discontinuities which lead to ambiguities in segmentation. Recently, Chan, Lam, and Zhu [52] proposed new adaptive thresholding method using variational theory for image segmentation, they can achieve better performances.

Region growing segmentation method [53] is probably less common than edge detection as a low level approach, but it is applicable in multi-dimensional cases and in noisy environments. Region growing algorithm starts by choosing a set of *"seed"* pixels and compare it with neighboring pixels. Then choose criteria for region growing, and region is *grown* from the seed pixel by adding in neighboring pixels that are similar. This process is continued until all pixels belong to some region. A new region growing method using novel similarity and discontinuity measures has been proposed by Hojjatoleslami and Kittler [54]. This method is more reliable and consistent than other region growing methods when the aim is the segmentation of bright objects from a dark background or vice versa. Region growing methods can be made less sensitive to noise than simple edge–based methods, but they may become extremely computationally complex for even simple rules [55]. Region growing method can also be applied to 3–D medical image segmentation [56].

The *morphological watersheds* segmentation method is another early-developed approach which embodies many of the concepts of the three approaches we just discussed above. It can produce more stable segmentation results including continuous segmentation boundaries [2]. Based on the multiscale behavior of gradient watershed hierarchies, an image segmentation algorithm has also been proposed by Gauch [57], it can be used for automatic and interactive image segmentation. Another method for

segmenting images using multiscale morphology was also developed very recently [58]. The PDE and morphology method can also be used for image segmentation [59].

In addition to the above deterministic methods for image segmentation, *stochastic modeling* is another main image analysis approach. *Stochastic modeling* is widely based on Markov random field (MRF) theory [60]. In these methods, image properties described in terms of probability distributions and some prior knowledge about the underlying image defined stochastically (*i.e.* prior probability model). Since there is one-to-one correspondence between Markov random field and Gibbs distribution function, one can formulate an *a posteriori* probability distribution based on observations and prior probability models [61]. Many MRF–based image segmentation approaches have also been proposed recently [62]–[64].

Another popular deterministic approach to the image segmentation problem is via *active contours* (also called snakes, curve evolution). The underling principle in these active contour methods is to define a simple, closed curve C_0 and deform (shrink or split) it towards the object boundaries. This idea was originally introduced by Kass, Witkin, and Terzopoulos [7] via an energy functional. These models are *active*, because they are always minimizing their energy functional and therefore exhibit dynamical behavior. More generally, the curve evolution is governed by the PDE as follows [65], [66]:

$$\frac{\partial \vec{C}}{\partial t} = F(\kappa, t)\vec{N}$$

where $\vec{N}$ is the normal to the curve, and $F(\kappa, t)$ is the velocity function which usually depends on the curvature κ of the curve. In order to stop the evolving curve at boundaries within an image, an edge strength function which depends on the local gradient of an image is used to control the evolution of the curve. These active contour models using the

gradient of the image for the stopping criteria are called as boundary–based models. We will examine some of them (snakes and the geodesic active contour models) in the next subsection.

3.1.2 Edge–Based Active Contours

The basic principle of the active contour models is the curve evolution and geometric flow; different models are defined by choosing different energy functional. An energy functional is Minimized so that its (local) minimum is obtained at the boundary of the object. The aim of these approaches is to detect contours of objects based on a sharp change of the image intensity $u(x, y)$ between the background and the objects. Usually we choose the gradient of an image to measure this sharp change of $u(x, y)$. Therefore the magnitude of the gradient of $u(x, y)$ is high across the edges, and we may choose $|\nabla u(x, y)|$ (or a function of it) as a edge detector.

a) *The Snakes Model*

Snakes are planar deformable contours that are used to approximate the locations and shapes of object boundaries within an image [7], [67]. Geometrical snakes, including energy–minimizing and dynamic snakes, are explicit, parametric contours embedded in the image plane. In the energy–minimizing snake approach, the final shape of the contour corresponds to the minimum of the energy of the contour, which is represented by the following energy functional [7]:

$$J(C) = \int_0^1 \alpha(s)\,|\,C'(s)\,|^2\,ds + \int_0^1 \beta(s)\,|\,C''(s)\,|^2\,ds + \lambda \int_0^1 g^2\big(|\,\nabla u(C(s))\,|\big)\,ds\,, \qquad (3.1)$$

where s is a point on the snake $C(s)$. The first two terms in (3.1) represent the internal energy and control the deformation of the contour. There are two non–negative parameter

functions: $\alpha(s)$ controls the "tension" of the contour while $\beta(s)$ controls its "rigidity". For example, increasing $\alpha(s)$ tends to eliminate extraneous loops and ripples by reducing the length of the snake. Increasing $\beta(s)$ makes the snake smoother and less flexible. The last term in (3.1) is the external energy and attracts the contour toward the intensity edges within an image. In general, the external energy is based on image attributes, and the object of interest, such as lines, edges, or other image features of interest. Here we use the edge–detector function $g(s)$ to define the external energy. $g(s)$ is regular monotonic decreasing and $g(0) = 1$, while $\lim_{s\to\infty} g(s) = 0$. A typical choice of g is $g(s) = 1/(1+s^2)$.

Minimizing the energy functional (3.1), we can obtain the *Euler–Lagrange* equation for the evolution of the snake as follows [67]:

$$-\frac{\partial}{\partial s}\left(\alpha\frac{\partial C}{\partial s}\right)+\frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 C}{\partial s^2}\right)+\lambda\nabla\left(g^2\left(|\nabla u(C(s))|\right)\right)=0. \tag{3.2}$$

The first two terms represent respectively the internal stretching and bending forces, and the third term is the external force that couples the snake to the image data.

The *dynamic snakes* can be represented by introducing a time–dependent contour $C(s, t)$ along with a *mass density* $\mu(s)$ and *damping density* $\gamma(s)$. Again minimizing the energy functional (3.1), the Euler–Lagrange equation of motion for a snake is given by [67]:

$$\mu\frac{\partial^2 C}{\partial t^2}+\gamma\frac{\partial C}{\partial t}=\frac{\partial}{\partial s}\left(\alpha\frac{\partial C}{\partial s}\right)-\frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 C}{\partial s^2}\right)-\lambda\nabla\left(g^2\left(|\nabla u(C(s))|\right)\right). \tag{3.3}$$

The left hand side of this partial differential equation represents inertial and damping forces.

The energy functional $J(C)$ in (3.1) is non-intrinsic since it depends on the parameterization of the curve C and is not directly related to the objects geometry [8]. We

may obtain different solutions by changing the parameterization within the same initial curve. And because of the regularity constraint, this model is not able to handle changes in the topology of the evolving contour directly. In fact, it is impossible to detect more than one object simultaneously [6].

The geodesic active contours model [8] which we present next, can overcome the above difficulties.

b) The Geodesic Active Contours Model

In the Kass et al. snakes model (3.1), if we set the rigidity term to zero that is $\beta = 0$, we obtain a particular case of the snakes model as (consider α as a constant),

$$J_1(C) = \alpha \int_0^1 |C'(s)|^2 \, ds + \lambda \int_0^1 g^2(|\nabla u(C(s))|) \, ds . \tag{3.4}$$

We can locate the boundaries of objects by minimizing the functional $J_1(C)$. But $J_1(C)$ is also not intrinsic because it still depends on the parameterization of C. So Caselles et al. proposed an intrinsic energy functional as follows [8], [68]:

$$J_2(C) = 2\sqrt{\lambda} \int_0^1 g(|\nabla u(C(s))|)|C'(s)| \, ds . \tag{3.5}$$

It has been shown that minimizing the energy functional $J_2(C)$ is equivalent to minimizing $J_1(C)$ [69]. Furthermore $J_2(C)$ is intrinsic because it does not depend on the parameterization of the curve [6]. The *Euler-Lagrange equation* that governs the evolution of the curve, can be obtained as [8],

$$\frac{\partial C(t)}{\partial t} = \kappa g(u)\vec{N} - (\nabla g(u) \cdot \vec{N})\vec{N} , \tag{3.6}$$

where κ is the Euclidean curvature, $\vec{N}$ is the unit inward normal. As already discussed in [6], the Euler-Lagrange equation (3.6) associated with J_2 can be formulated by using the *level set methods* that make the numerical implementation much easier, while it is not the

case for J_1. The level set techniques have shown great promise for their speed and ability to handle changes in topology of the evolving curve and make the implementations more efficient and robust.

In both the snakes model and the geodesic active contours model, we minimize a variational energy functional to get the curve motion equation, and an edge detector function (depends on ∇u_0) is used to stop the evolving curve on the edges of the object. Another variational energy functional approach is to find an optimal piecewise smooth approximation u of an original image u_0, and a set of boundaries C, such that u varies smoothly within the homogeneous regions and discontinuously or rapidly across C. This is achieved by minimizing the Mumford–Shah energy functional [3]. We will examine the Mumford and Shah model and its properties in the next subsection.

3.1.3 The Mumford and Shah Model

Let Ω be a bounded open set of R^2, and $u_0(x, y)$ be an initial image. The basic idea of the Mumford and Shah model [3] is to find a pair (u, C) for a given image u_0, such that u is a nearly piecewise smooth approximation of u_0, and C is a set of edges between regions within the image. In order to solve this image segmentation problem, Mumford and Shah proposed the following energy functional:

$$F(u,C) = \int_{\Omega} |u - u_0|^2 \, dxdy + \mu \int_{\Omega \setminus C} |\nabla u|^2 \, dxdy + \nu \cdot Length(C), \tag{3.7}$$

where μ and ν are nonnegative constants, Ω bounds an open set of R^2 (image domain), the curve $C \subset \Omega$, and $\{x, y\} \in \Omega$. $u(x, y)$ is the piecewise smooth function approximate to u_0 with discontinuities *only* along the boundaries C. To solve the Mumford–Shah problem we need to minimize the energy functional $F(u, C)$ with respect to u and C. The authors

made the following conjecture in their original paper [3] for the existence of a minimizer in this minimization problem:

Conjecture: *There exists a minimizer of F such that the edges (the discontinuity set C) are the union of a finite set of $K^{1,1}$ embedded curves. Moreover, each curve may end either as a crack tip (a free extremity, i.e., C looks like a half–line) or in triple junction, that is, three curves meeting at their endpoints with $2\pi/3$ angle between each pair.*

It is important to notice that the Mumford–Shah functional is a concise mathematical definition of the image segmentation problem. Bonnet made important progress in the existence of a minimizer of the functional (3.7), and proved the following results:

***Theorem* 1** [6], [70]: *If (u, C) is a minimizer of F such that C is connected, then (u, C) is one of the following:*

i) *C is empty and u is constant.*

ii) *C is a straight line defining two half-planes and u is constant on each half plane.*

iii) *C is the union of three half lines with $2\pi/3$ angles and u is constant on each sector.*

iv) *In a polar set of coordinates (r, θ), $u(r,\theta) = \sqrt{2/\pi}\sqrt{r}\cos(\theta/2)$ for $\theta \in [0, 2\pi[$, and C is the half–axis $\theta = 0$ (a crack-tip).*

***Theorem* 2** [6], [70]: *Every isolated connected component of C is the union of a finite set of K^1–arcs. These arcs are $K^{1,1}$ away from crack–tips and can merge through triple junctions with $2\pi/3$ angles.*

We notice that Theorem 2 does not allow a minimizer to have an infinite number of arbitrarily small pieces connected to each other. Actually, the Mumford and Shah

conjecture remains an open question in the most general setting [6]. Based on the quasi–Newton minimizing algorithm, the numerical minimization techniques of the Mumford–Shah functional have been developed recently [71]. The difficulties in studying the Mumford–Shah functional $F(u, C)$ are that it involves two unknowns u and C of different natures: u is a function in an N–dimensional space ($N = 2$ in our consideration), while C is an $(N-1)$–dimensional set. The other difficulty is that the functional $F(u, C)$ is not convex, and may have numerous local minima. Therefore, it is not easy to minimize the Mumford–Shah functional (3.7) in practice, there are some alternative solutions to this problem, such as, the elliptic approximation to the weak formulation of the Mumford–Shah functional method [6], the active contours without edges [9]–[13], and the curve evolution based approach [5]. The characteristics of these methods are:

i) An active contour deforms within an image and stops when it reaches the boundary of regions. So the single active contour cannot detect a region if it is inside another region.

ii) Because the level set method is used in these approaches, we always have closed curves.

Generally, the Mumford–Shah energy functional model can perform curve evolution, segmentation, and smoothing in a common way [72]. We present recent works that consider these characteristics in the next subsection.

3.1.4 Active Contour without Edges and the Level Set Methods

Many active contour models use the level set method to represent the evolving curve, because the level set method allows automatic topology changes, such as merging and breaking of curves. Recently new active contours with level set methods have been

proposed for images segmentation such as, edge based models [7], [8], [67], [68], [73]–[76], region based techniques [77], [78], curve evolution approaches [5], [79], [80], shape based models [81]–[84], color imagery [11], and active contours without edges [10]–[14]. Among these methods, we would like to examine those closely related with the work of this thesis.

a) The Chan–Vese piecewise smooth model

From the general form of the Mumford–Shah functional (3.7), if we consider that there is a closed curve (active contour) C in the image domain Ω, Ω is partitioned into R and $\overline{R}$ corresponding to the image sub-domains inside and outside the curve C respectively. Then minimizing (3.7) becomes the minimization of the following problem [4], [5]:

$$F(u_1, u_2, C) = \int_{inside\,(C)} |u_1 - u_0|^2 \, dxdy + \mu \int_{inside\,(C)} |\nabla u_1|^2 \, dxdy$$

$$+ \int_{outside\,(C)} |u_2 - u_0|^2 \, dxdy + \mu \int_{outside\,(C)} |\nabla u_2|^2 dxdy + \nu \cdot Length(C). \qquad (3.8)$$

where u_1 and u_2 are the smooth function approximating the image function u_0 inside and outside the curve respectively, and μ and ν are constants. We apply the level set method to this model by replacing the unknown curve $C(t)$ by the level set function $\phi(x, y, t)$, and consider that $\phi(x, y, t) > 0$ if (x, y) is inside the curve C, $\phi(x, y, t) < 0$ if (x, y) is outside curve C and $\phi(x, y, t) = 0$ if (x, y) is located on the curve C. Minimizing the functional $F(u_1, u_2, \phi)$ with respect to u_1, u_2 and ϕ, we obtain the equations for u_1, u_2 and ϕ as the following:

$$u_1 - u_0 = \mu \nabla^2 u_1 \quad \text{inside } C, \; \frac{\partial u_1}{\partial \vec{n}} = 0 \text{ on } C \qquad (3.9)$$

$$u_2 - u_0 = \mu \nabla^2 u_2 \text{ outside } C, \quad \frac{\partial u_2}{\partial \vec{n}} = 0 \text{ on } C \tag{3.10}$$

$$\frac{\partial \phi}{\partial t} = \delta(\phi)\left[\nu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right) - |u_1 - u_0|^2 - \mu |\nabla u_1|^2 + |u_2 - u_0|^2 + \mu |\nabla u_2|^2 \right], \tag{3.11}$$

$\delta(\phi)$ is the Dirac δ function. The image functions $u_1(x, y)$ and $u_2(x, y)$ are obtained by solving the damped Poisson equations (3.9) and (3.10) for any given curve C. This is the piecewise smooth case of the Chan–Vese model [4]. Very similar idea has been also developed independently by Tsai, Yezzi and Willsky [5]. The smoothing and denoising effect on the image u_0 comes from solving the PDEs for u_1 and u_2, which are inside and outside the curve respectively. Therefore diffusion filtering only happens within different homogeneous regions, but not across the boundaries. The smoothing approach is very similar to the idea of "anisotropic diffusion" [15]–[17], [21]–[25].

b) The piecewise constant segmentation methods

There are two piecewise constant image segmentation approaches we would like to present here, namely, the active contour model with level set formulation developed by Chan and Vese in [9], [4], and the direct energy computation method proposed by Song and Chan in [85].

i) The Chan–Vese model: The piecewise constant active contour model proposed by Chan and Vese [9] is a particular case of the Mumford–Shah model. It minimizes the following energy functional

$$F(c_i, C_i) = \sum_i \int_{\Omega_i} (u_0(x, y) - c_i)^2 dxdy + \nu \cdot Length(C), \tag{3.12}$$

where c_i is the average value of $u_0(x, y)$ in each connected region Ω_i, and $\Omega = \cup_i \Omega_i \cup C$. ν is a positive constant. Using the Heaviside function $H(\phi)$ this energy functional can be

represented by the level set approach with $C(t)$ corresponding to the zero level set of the level set function, i.e. $\phi(x, y, t) = 0$. The length of C can be expressed by $|C| = \int_{\Omega} |\nabla H(\phi)| dxdy$. For n–phase image, $m = \log_2 n$ level set functions $\phi_i(x, y, t)$ are needed. Then the level set formulation of the energy functional (3.12) can be written as follows,

$$F(c,\Phi) = \sum_{1 \le I \le n} \int_{\Omega} (u_0(x,y) - c_I)^2 \chi_I dxdy + \nu \cdot \sum_{1 \le i \le m} \int_{\Omega} |\nabla H(\phi_i)| dxdy, \tag{3.13}$$

where $c = (c_1, \ldots, c_I, \ldots, c_n)$, $\Phi = (\phi_1, \ldots, \phi_I, \ldots, \phi_m)$, c_I is the average of u_0 in region I, and χ_I is the characteristic function in region I.

For example, in 2–phase segmentation ($n = 2$, $m = 1$) case, we can use only one level set function ϕ to represent the two-phase energy functional as,

$$F_2(\phi, c_1, c_2) = \nu \int_{\Omega} |\nabla H(\phi)| dxdy + \int_{\Omega} |u_0 - c_1|^2 H(\phi) dxdy + \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi)) dxdy. \tag{3.14}$$

It is to minimize the functional (3.14) with respect to c_1, c_2 and ϕ to obtain the equations for c_1, c_2, and the curve $C(t)$ respectively:

$$c_1(\phi) = \frac{\int_{\Omega} u_0 H(\phi) dxdy}{\int_{\Omega} H(\phi) dxdy}, \qquad c_2(\phi) = \frac{\int_{\Omega} u_0 (1 - H(\phi)) dxdy}{\int_{\Omega} (1 - H(\phi)) dxdy}, \tag{3.15}$$

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\nu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - (u_0 - c_1)^2 + (u_0 - c_2)^2 \right]. \tag{3.16}$$

The segmented image therefore is a two–phase image, given by $u(x, y) = c_1 H(\phi) + c_2 [1 - H(\phi)]$. After solving these equations we can obtain the information about c_1, c_2, and C, and we expect that u is the best approximation of u_0 with two values c_1, c_2 and one

edge C. The curve C is the boundary between the sets $\{u = c_1\}$ and $\{u = c_2\}$, therefore we segment the original image u_0 into two parts $\{u = c_1\}$ and $\{u = c_2\}$.

ii) Direct energy computation method: As we already mentioned, the main idea of variational active contour model is to minimize the energy functional, and this model can be formulated by the level set method. We can locate the boundaries by finding the level set function ϕ_t, this can be done by solving the corresponding PDE (3.16). However, for the active contour segmentation model, we do not really need the values of the level set function ϕ_t, but only its sign. So it is possible to solve the minimization problem by directly computing the energy F. Based on this observation, we present the direct energy computation algorithm introduced by Song and Chan in [85] to solve the segmentation problem as follows:

Step 1: Construct an initial partition. The initial curve corresponding to $\phi = 0$ divides the image into two parts, one part for $\phi > 0$, and another part for $\phi < 0$. Then compute the value of the energy difference ΔF according to the sign of ϕ.

Step 2: Advance. For each point (x, y) in the image, if the energy F decreases when we change $\phi(x, y)$ to $-\phi(x, y)$ (i.e. moving the point from inside to outside of the curve or vice versa), then update this point by $\phi(x, y) = -\phi(x, y)$, otherwise, $\phi(x, y)$ remains unchanged.

Step 3: Repeat step 2 until the energy F remains unchanged.

The main advantage of the direct energy computation algorithm is that we do not need to solve the level set PDE for the curve evolution, thus it improves the computational speed, especially for image with large size.

For 2–phase segmentation, given an initial curve that separates the image into two parts, let $\phi = 1$ for the region inside the curve and $\phi = -1$ outside the curve. Let c_1 and c_2 be the averages in regions for $\phi = 1$ and $\phi = -1$ respectively, m and n are the numbers of pixels for $\phi = 1$ and $\phi = -1$, and let F_O be the energy at that moment. Assume that the current pixel $p(x, y) \in \{\phi = 1\}$ with the intensity value of $I(x, y)$. If we change $p(x, y)$ from region $\{\phi = 1\}$ to region $\{\phi = -1\}$, the number of pixels in $\{\phi = 1\}$ becomes $m-1$, while the number of pixels in $\{\phi = -1\}$ becomes $n+1$, the energy averages in regions $\{\phi = 1\}$ and $\{\phi = -1\}$ are also changed. Let F_N be the new energy after we change $p(x, y)$ from $\{\phi = 1\}$ to $\{\phi = -1\}$, the energy difference between the new energy F_N and the old energy F_O can be computed as [85],

$$\Delta F_{12} = F_N - F_O = \left(I(x,y) - c_2\right)^2 \frac{n}{n+1} - \left(I(x,y) - c_1\right)^2 \frac{m}{m-1}. \tag{3.17}$$

If $\Delta F_{12} < 0$, change the value of $\phi(x, y)$ from 1 to −1. Similarly for the $\phi = -1$ case.

More generally, for image segmentation the deterministic edge detection–based, region–based, active–contour–based, and stochastic models are subsets of the general problem of variational functional minimization. This is our motivation for choosing the Mumford–Shah variational energy functional as the starting point for our segmentation method.

3.2 PDE–Based Image Smoothing Methods

3.2.1 Overview

The PDE–based, nonlinear anisotropic diffusion techniques are an effective way to smooth and denoise images. The smoothed image can be obtained by the solution $u(x, y)$

of the diffusion equation at a specific time t with initial conditions $u(x, y, t=0) = u_0(x, y)$, where $u_0(x, y)$ is an *original* noisy image. The general formula of a diffusion equation can be written as,

$$\frac{\partial u(x,y)}{\partial t} = \nabla \cdot (\mathbf{D}(x,y) \cdot \nabla u(x,y)), \tag{3.18}$$

$$u(x,y,t=0) = u_0(x,y),$$

where D is the diffusivity. According to the property of the divergence operation, the diffusivity can be a scalar function (or constant) or a tensor (or tensor-valued function).

The simplest diffusion filter is the linear isotropic diffusion filter, where the diffusivity is a scalar constant. It smoothes the noise in an image and blurs the edges of objects within the image as well. In order to avoid blurring of edges, the nonlinear isotropic diffusion [86] uses a scalar function of gradient ∇u_σ instead of a constant diffusivity, where u_σ is the regularized image by convolving the original noisy image u_0(x, y) with a Gaussian filter $K_\sigma(x, y)$, which is $u_\sigma \equiv u_0 * K_\sigma$ and $D = g(|\nabla u_\sigma|)$. This diffusivity function satisfies g(0) =1 and g(s) = 0 for $s \to \infty$. Therefore it behaves as linear diffusion in the interior of a region ($|\nabla u_\sigma| \to 0$) and inhibits diffusion at strong edges ($|\nabla u_\sigma| \to \infty$). The nonlinear isotropic diffusion can avoid blurring of edges, but it cannot eliminate noise at edges [25], [86].

3.2.2 Anisotropic Diffusion Method for Image Smoothing

The anisotropic diffusion technique, first introduced in [15] and further developed in [16], [17], takes into account both the modulus of the gradient $|\nabla u_\sigma|$ and its direction. Here $D(x, y)$ is generally a symmetric positive definite diffusion tensor. It smoothes the image within the homogeneous regions and/or probably along the edges, but not across

the boundaries of such regions. Below we present the anisotropic diffusion techniques developed by Weickert [21]–[25].

The basic diffusion equation is given by equation (3.18). In the anisotropic diffusion approach, the diffusivity $D(x, y)$ is chosen as a tensor function that contains information of $|\nabla u_\sigma|$ and the direction of $\vec{\nabla} u_\sigma$ as well. The key point for this anisotropic diffusion approach is how to construct the diffusion tensor D. An easy way to construct D is from the direct product of ∇u_σ:

$$D = \nabla u_\sigma \otimes \nabla u_\sigma = \nabla u_\sigma (\nabla u_\sigma)^T = \begin{pmatrix} u_x \\ u_y \end{pmatrix} \begin{pmatrix} u_x & u_y \end{pmatrix} = \begin{pmatrix} u_x^2 & u_x u_y \\ u_x u_y & u_y^2 \end{pmatrix} \tag{3.19}$$

In order to construct the diffusion tensor D for applications, we introduce the structural tensor which is a symmetric, positive semi-definite matrix.

a) The Structural Tensor:

$$J_\rho(\nabla u_\sigma) = K_\rho * (\nabla u_\sigma \otimes \nabla u_\sigma) \qquad (\rho \geq 0), \tag{3.20}$$

where K_ρ is the Gaussian filter. Since $J_\rho(\nabla u_\sigma)$ is a symmetric matrix, it has the following form:

$$\begin{pmatrix} j_{11} & j_{12} \\ j_{12} & j_{22} \end{pmatrix}$$

Assume that the matrix $J_\rho(\nabla u_\sigma)$ has an eigenvalue μ and the corresponding eigenvector $\vec{v} = \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix}$, thus we have the following equation:

$$\begin{pmatrix} j_{11} & j_{12} \\ j_{12} & j_{22} \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} = \mu \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix}. \tag{3.21}$$

It is equivalent to

$$\begin{pmatrix} j_{11}-\mu & j_{12} \\ j_{12} & j_{22}-\mu \end{pmatrix}\begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix}=0. \tag{3.22}$$

The eigenvalues of the matrix $J_\rho(\nabla u_\sigma)$ can be determined by the determinant of the left-hand side 2×2 matrix in equation (3.22), which are:

$$\mu_{1,2}=\frac{1}{2}\left[j_{11}+j_{22}\pm\sqrt{(j_{11}-j_{22})^2+4j_{12}^2}\right], \tag{3.23}$$

where μ_1 corresponds to '+' sign in (3.23). Since the matrix $J_\rho(\nabla u_\sigma)$ has two eigenvalues μ_1 and μ_2, there are two corresponding eigenvectors $\vec{v}_1$ and $\vec{v}_2$ respectively.

When $\mu = \mu_1$, from equation (3.21) and the orthonormal condition, we obtain the eigenvector $\vec{v}_1=\begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix}$, with,

$$\varphi_1=\frac{2j_{12}}{\sqrt{\left(j_{22}-j_{11}+\sqrt{(j_{11}-j_{22})^2+4j_{12}^2}\right)^2+4j_{12}^2}},$$

$$\varphi_2=\frac{j_{22}-j_{11}+\sqrt{(j_{11}-j_{22})^2+4j_{12}^2}}{\sqrt{\left(j_{22}-j_{11}+\sqrt{(j_{11}-j_{22})^2+4j_{12}^2}\right)^2+4j_{12}^2}}.$$

When $\mu = \mu_2$, similarly, we get the eigenvector $\vec{v}_2=\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}$, where,

$$\xi_1=\frac{2j_{12}}{\sqrt{\left(j_{22}-j_{11}-\sqrt{(j_{11}-j_{22})^2+4j_{12}^2}\right)^2+4j_{12}^2}},$$

$$\xi_2 = \frac{j_{22} - j_{11} - \sqrt{(j_{11} - j_{22})^2 + 4j_{12}^2}}{\sqrt{\left(j_{22} - j_{11} - \sqrt{(j_{11} - j_{22})^2 + 4j_{12}^2}\right)^2 + 4j_{12}^2}}.$$

It is easy to verify that the eigenvectors $\vec{v}_1$ and $\vec{v}_2$ are orthogonal, since

$$\vec{v}_1 \cdot \vec{v}_2^T = \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} \cdot (\xi_1 \quad \xi_2) = \varphi_1\xi_1 + \varphi_2\xi_2 = 0.$$

b) A Special Case

If the structural tensor has the form of equation (3.19), following the discussions above, we obtain its eigenvalues as,

$$\mu_1 = u_x^2 + u_y^2 = |\nabla u|^2, \qquad \mu_2 = 0$$

The corresponding eigenvectors are:

$$\vec{v}_1 = \begin{pmatrix} u_x/|\nabla u| \\ u_y/|\nabla u| \end{pmatrix}, \text{ parallel to } \vec{\nabla} u \; ; \; \vec{v}_2 = \begin{pmatrix} u_y/|\nabla u| \\ u_x/|\nabla u| \end{pmatrix}, \text{ orthogonal to } \vec{\nabla} u .$$

Based on the above discussions, now we can present how to construct the diffusion tensor $\boldsymbol{D}(J_\rho)$ as follows.

c) Construct the Diffusion Tensor

Assume that the diffusion tensor $\boldsymbol{D}$ we would like to construct has the form:

$$\mathbf{D} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}.$$

The elements of $\boldsymbol{D}$ will be determined in the following way: we choose the orthonormal basis of the eigenvectors of $J_\rho(\nabla u_\sigma)$, $\vec{v}_1$ and $\vec{v}_2$ as the eigenvectors of $\boldsymbol{D}$, while the eigenvalues of $\boldsymbol{D}$ denoted by λ_1 and λ_2 are different from the eigenvalues μ_1 and

μ_2 of $J_\rho(\nabla u_\sigma)$. The choice of λ_1 and λ_2 depends on the desired goal of application. Since $\vec{v}_1$ and $\vec{v}_2$ are orthonormal vectors, $\boldsymbol{D}$ can be constructed as,

$$\mathbf{D}=\begin{pmatrix} a & b \\ b & c \end{pmatrix}=\begin{pmatrix} \vec{v}_1 & \vec{v}_2 \end{pmatrix}\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}\begin{pmatrix} \vec{v}_1^T \\ \vec{v}_2^T \end{pmatrix}=\begin{pmatrix} \varphi_1 & \xi_1 \\ \varphi_2 & \xi_2 \end{pmatrix}\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}\begin{pmatrix} \varphi_1 & \varphi_2 \\ \xi_1 & \xi_2 \end{pmatrix}$$

$$=\begin{pmatrix} \lambda_1\varphi_1^2+\lambda_2\xi_1^2 & \lambda_1\varphi_1\varphi_2+\lambda_2\xi_1\xi_2 \\ \lambda_1\varphi_1\varphi_2+\lambda_2\xi_1\xi_2 & \lambda_1\varphi_2^2+\lambda_2\xi_2^2 \end{pmatrix}. \tag{3.24}$$

It is easy to verify that if $\boldsymbol{D}$ has the form as equation (3.24), $\vec{v}_1$ and $\vec{v}_2$ are the eigenvectors of $\boldsymbol{D}$ with the eigenvalues λ_1 and λ_2 respectively, then:

$$\mathbf{D}\begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix}=\lambda_1\begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix},\ \text{and}\ \mathbf{D}\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}=\lambda_2\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}.$$

Weickert proposed two different ways to choose the diffusion tensor $\boldsymbol{D}$ for different diffusion goals, namely the edge-enhancing and coherence-enhancing anisotropic diffusions.

a) Edge-enhancing anisotropic diffusion [21]: The goal of this type of diffusion filter is to smooth within the homogenous region and preserve edges. Thus we should reduce the diffusivity λ_1 perpendicular to edges. The goal can be accomplished by choosing the eigenvalues of λ_1 and λ_2 as follows:

$$\lambda_1=\begin{cases} 1 & \text{if} \quad \mu_1=0 \\ 1-\exp\left(-\dfrac{3.315}{\mu_1^4}\right) & \text{otherwise} \end{cases}$$

$$\lambda_2=1.$$

b) Coherence-enhancing anisotropic diffusion [22]: If we want to enhance flow-like structures and close broken lines, we should smooth preferably along the coherence

direction $\vec{v}_2$ with the diffusivity λ_2 that increases with respect to the coherence $(\mu_1 - \mu_2)^2$. This can be achieved by the following choice of the eigenvalues of λ_1 and λ_2:

$$\lambda_1 = \alpha,$$

$$\lambda_2 = \begin{cases} \alpha & \text{if} \quad \mu_1 = \mu_2, \\ \alpha + (1-\alpha)\exp\left(\dfrac{-1}{(\mu_1 - \mu_2)^2}\right) & \text{otherwise,} \end{cases}$$

where α is a small positive parameter $\in$ (0, 1).

Among PDE diffusion approaches for image denoising, the anisotropic diffusion approach gives the highest performance. The fundamental idea of the anisotropic diffusion filtering is to smooth the images isotropically within a region, while diffusing in an anisotropic way along edges of the regions (*e.g.* diffusion along the edges but not across the edges). The most important advantage of the anisotropic diffusion technique is that it selectively smoothes an image while preserving and relatively enhancing the edges of regions.

In the methods mentioned above, the edge detector is based on the gradient of the convolved image ∇u_σ. In the next section, we will present our method to detect edges and to anisotropically smooth noisy images without using the image gradients.

Chapter 4

The Hierarchical Image Segmentation Method

In order to segment images with complicated features properly, we need multiphase segmentation techniques by employing multiple level set functions. In this chapter we first examine the Chan and Vese multiphase segmentation model, then present our hierarchical multiphase piecewise method which has many advantages compared with the Chan–Vese model.

4.1 The Multiphase Image Segmentation Method

As we have already mentioned in chapter 3, the Chan and Vese 2–phase active contour model [9] generalizes the Mumford–Shah functional model with *one* level set function. The interior contours in simple binary images can be detected successfully, but there are some important features such as triple junction points in general gray level images that cannot be detected in this model. This is because we can only represent an image in two phases (binary segmentation) by using only *one* level set function (see Fig. 4.2 below). For images with more complicated features, usually multiple level set functions are

needed in order to obtain proper segmentations. We now present details of the multiphase segmentation method.

4.1.1 Motivations

The Chan and Vese active contours model with one level set function [9] (as we have discussed in chapter 3) detects the contours within an image by solving the contour evolution equation represented by the level set function ϕ (3.15), and (3.16) under the Neumann boundary condition, $\frac{\partial \phi}{\partial n} = 0$ on $\partial\Omega$. In the numerical implementation, we use the regularized version of the Heaviside function $H(\phi)$ and the Dirac delta function $\delta(\phi)$, defined by,

$$H_\varepsilon(\phi) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\phi}{\varepsilon}\right)\right), \qquad \delta_\varepsilon(\phi) = H'_\varepsilon(\phi) = \frac{1}{\pi}\cdot\frac{\varepsilon}{\varepsilon^2 + \phi^2}.$$

As $\varepsilon \to 0$, both approximations of H_ε and δ_ε converge to their original definitions,

$$H(\phi) = \begin{cases} 1, & \text{if } \phi \geq 0 \\ 0, & \text{if } \phi < 0 \end{cases}, \qquad \delta(\phi) = \begin{cases} \frac{1}{2\varepsilon}, & \text{if } -\varepsilon \leq \phi \leq \varepsilon \\ 0, & \text{otherwise} \end{cases}.$$

From the viewpoint of numerical implementation, this algorithm works as the following steps:

1. Initialize ϕ_0 from an initial curve $C(0) \leftrightarrow \phi(x, y, 0) = \phi_0$ ($n = 0$).
2. Compute $c_1(\phi_n)$ and $c_2(\phi_n)$ using the equations (3.15), and calculate the length term $length\{\phi_n = 0\} = Length(\phi_n)$.
3. Solve the PDE for ϕ to obtain ϕ_{n+1} from (3.16).
4. Reinitialize ϕ_n at each iteration step.
5. Check if the solution is stationary near the boundary. If not, $n=n+1$ and repeat.

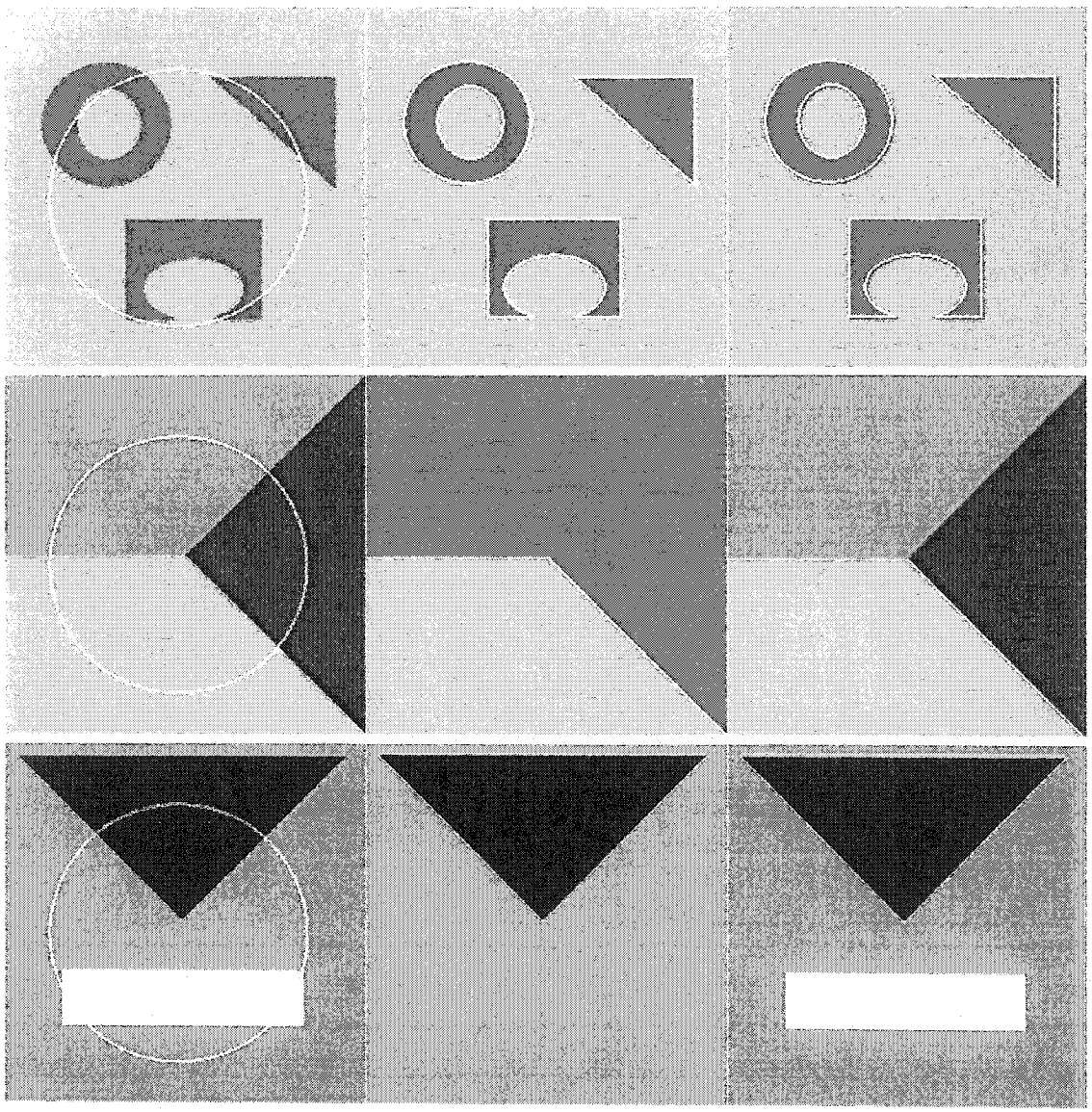


Fig. 4.1. Segmentation results using 2–phase segmentation algorithm. The far left column shows the original synthetic images with initial curves. The middle column contains the segmenting results by using piecewise constant approach with the segmenting curves superimposed on them. The right column shows the results using piecewise smooth representation with the segmenting curves superimposed on them.

This algorithm can automatically detect different shapes, convexities, and interior contours as already discussed in [9]. It works successfully for two–phase images, but it cannot detect the triple junction point and cannot represent an image with more than two

phases. We show an example in Fig. 4.1 with three synthetic images for these cases. We choose $\mu = 1$, $\nu = 100$, and $\varepsilon = 1$ in our numerical implementations.

Fig. 4.1 shows that, with the use of *one* level set function, we can segment an image into two regions, and we get the successful segmentations of a two–phase image which is shown in the first row. The middle column of Fig. 4.1 clearly indicates the segmented results in two parts by using a piecewise constant representation; and the white curves are the boundaries between these regions. Although we can show all parts within an image by using piecewise smooth representation (see the far right column), but there are still two regions. For example, the image in the second row of the far right column, the white rectangle and the background belong to one region (there is no boundary curve between them) while the black triangle belongs to another region. In order to overcome these problems, we generalize the active contour model to segment images with more than two regions by employing multiple level set functions. The main differences between one level set function approach and multiple (for example 2) level set functions approach are shown in Fig. 4.2. With one level set function, we have one curve (active contour) that partitions the image domain into two regions (see Fig. 4.2(a)). In the *two* level set functions approach (Fig. 4.2(b)), we have *two* initial curves (active contours) and these two contours evolve during the segmenting process. The two contours associated with two level set functions ϕ_1 and ϕ_2, partition the image domain up to 4 regions. Similarly, we can use 3 level set functions to partition an image domain up to 8 regions, and so on. Therefore we expect to get more detailed segmentations of an image by using multiple level set functions [4], [87].

4.1.2 The Multiphase Segmentation Models

We use 4–phase piecewise segmentation method as an example to illustrate the basic idea of the multiphase segmentation models. We first examine how to extend the Chan–Vase 2–phase active contour model to the 4–phase segmentation approach, then we generalize the direct energy computation method to segment images up to 4 regions based on the same idea of the Chan–Vase multiphase segmentation method.

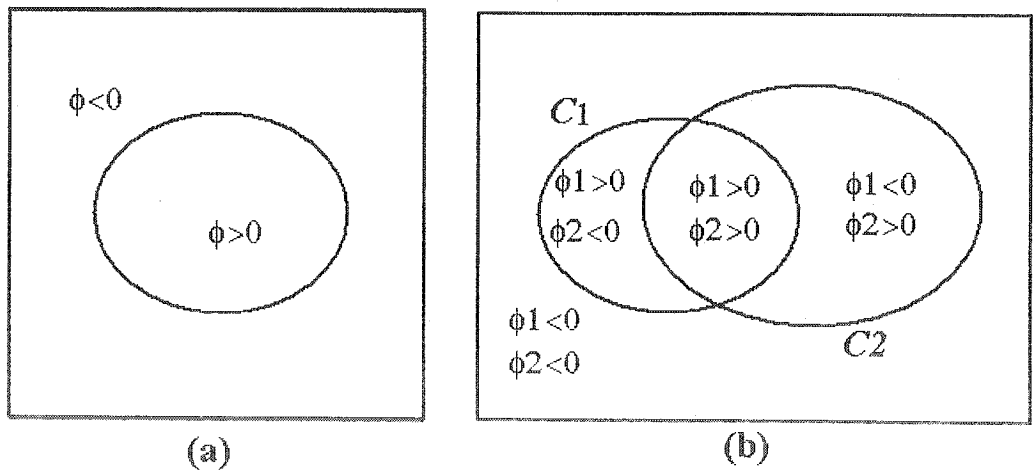


Fig. 4.2. Regions divided by one level set function ϕ and two level set functions ϕ_1 and ϕ_2. (a) One curve given by ϕ separates an image domain into 2 regions. (b) Two curves given by ϕ_1 and ϕ_2 partition the image domain into 4 regions.

A. The Chan and Vase Multiphase Segmentation Method

For the 4–phase segmentation case, two level set functions ϕ_1 and ϕ_2 are needed. Let $\Phi=(\phi_1, \phi_2)$. From the energy functional (3.12) proposed by Chan and Vese [88] based on the Mumford–Shah model and using two Heavside functions $H(\phi_1)$ and $H(\phi_2)$ associated with the level set functions ϕ_1 and ϕ_2 respectively, we can obtain the level set formulation of the 4–phase energy function as follows,

$$F_4(c,\Phi) = \int_\Omega |u_0 - c_{11}|^2 H(\phi_1)H(\phi_2)dxdy + \int_\Omega |u_0 - c_{10}|^2 H(\phi_1)(1-H(\phi_2))dxdy$$

$$+ \int_\Omega |u_0 - c_{01}|^2 (1-H(\phi_1))H(\phi_2)dxdy + \int_\Omega |u_0 - c_{00}|^2 (1-H(\phi_1))(1-H(\phi_2))dxdy$$

$$+ \nu \int_\Omega |\nabla H(\phi_1)|\, dxdy + \nu \int_\Omega |\nabla H(\phi_2)|\, dxdy . \qquad (4.1)$$

Where $\mathbf{c} = \{c_{11}, c_{00}, c_{10}, c_{01}\}$ are the constants in each corresponding region, and the segmented image therefore is a 4–phase image, given by

$$u = c_{11}H(\phi_1)H(\phi_2) + c_{10}H(\phi_1)(1-H(\phi_2)) + c_{01}(1-H(\phi_1))H(\phi_2) + c_{00}(1-H(\phi_1))(1-H(\phi_2))$$

The Euler–Lagrange equations for c and Φ can be obtained respectively by minimizing the energy functional with respect to c and Φ. We obtain the equation to compute the value in each of the four regions as follows,

$$c_{11}(\Phi) = \frac{\int_\Omega u_0 H(\phi_1)H(\phi_2)\ dxdy}{\int_\Omega H(\phi_1)H(\phi_2)\ dxdy}, \qquad (4.2a)$$

$$c_{10}(\Phi) = \frac{\int_\Omega u_0 H(\phi_1)(1-H(\phi_2))\ dxdy}{\int_\Omega H(\phi_1)(1-H(\phi_2))\ dxdy}, \qquad (4.2b)$$

$$c_{01}(\Phi) = \frac{\int_\Omega u_0 (1-H(\phi_1))H(\phi_2)\ dxdy}{\int_\Omega (1-H(\phi_1))H(\phi_2)\ dxdy}, \qquad (4.2c)$$

$$c_{00}(\Phi) = \frac{\int_\Omega u_0 (1-H(\phi_1))(1-H(\phi_2))\ dxdy}{\int_\Omega (1-H(\phi_1))(1-H(\phi_2))\ dxdy}. \qquad (4.2c)$$

The curve evolution equations which are the *Euler–Lagrange* PDEs for the level set functions ϕ_1 and ϕ_2 are given by,

$$\frac{\partial \phi_1}{\partial t} = \delta(\phi_1)\{\nu\ \nabla \cdot \left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right)$$

$$-\left[\left((u_0-c_{11})^2-(u_0-c_{01})^2\right)H(\phi_2)+\left((u_0-c_{10})^2-(u_0-c_{00})^2\right)(1-H(\phi_2))\right]\Big\}, \quad (4.3a)$$

$$\frac{\partial\phi_2}{\partial t}=\delta(\phi_2)\Big\{\nu\ \nabla\cdot\left(\frac{\nabla\phi_2}{|\nabla\phi_2|}\right)$$

$$-\left[\left((u_0-c_{11})^2-(u_0-c_{10})^2\right)H(\phi_1)+\left((u_0-c_{01})^2-(u_0-c_{00})^2\right)(1-H(\phi_1))\right]\Big\}. \quad (4.3b)$$

Solving the above PDEs (4.3) for ϕ_1 and ϕ_2, we can obtain the evolution of the curves $C_1(t)$ and $C_2(t)$. The average values c_{11}, c_{00}, c_{10}, and c_{01} can be calculated from equations (4.2) for different $\phi_i(x, y, t)$ which are determined by the motion of curve $C_i(t)$ (since $C_i(t) \leftrightarrow \phi_i(t)$). The moving curves will stop at the boundaries of the objects within an image.

Although this algorithm works well and gets better performance than 2–phase approach for image segmentation, solving the coupled Euler–Lagrange equations (4.3a) and (4.3b) costs even more *CPU* time. The direct energy computation method gives a way to improve the computational speed [85] as we discussed in section 3.1.4 (page 30). We extend this approach to the 4–phase case as follows.

B. The 4–Phase Direct Energy Computation Method

Using two level set functions ϕ_1 and ϕ_2 we can also extend the direct energy computation method to the 4–phase segmentation case. When we use two curves, an image domain can be divided into 4 different regions, let c_{ij} and n_{ij} be the average value and the number of pixels in each corresponding region, thus the average value and the number of pixels in each region can be denoted as follows,

Region "11": $\{\phi_1 > 0 \text{ and } \phi_2 > 0\}$, c_{11}, n_{11};

Region "10": $\{\phi_1 > 0 \text{ and } \phi_2 < 0\}$, c_{10}, n_{10};

Region "01": $\{\phi_1 < 0 \text{ and } \phi_2 > 0\}$, c_{01}, n_{01};

Region "00": $\{\phi_1 < 0 \text{ and } \phi_2 < 0\}$, c_{00}, n_{00}.

Assume that a pixel p(x, y) with the intensity value $I(x, y)$ in the region {$\phi_1 > 0$ and $\phi_2 > 0$}, if we move this pixel out of its current region, it will go to one of the other three regions. So we need to test the energy differences:

$$\Delta F_{11\to 00} = \left(I(x,y) - c_{00}\right)^2 \frac{n_{00}}{n_{00}+1} - \left(I(x,y) - c_{11}\right)^2 \frac{n_{11}}{n_{11}-1}, \tag{4.4a}$$

$$\Delta F_{11\to 10} = \left(I(x,y) - c_{10}\right)^2 \frac{n_{10}}{n_{10}+1} - \left(I(x,y) - c_{11}\right)^2 \frac{n_{11}}{n_{11}-1}, \tag{4.4b}$$

$$\Delta F_{11\to 01} = \left(I(x,y) - c_{01}\right)^2 \frac{n_{01}}{n_{01}+1} - \left(I(x,y) - c_{11}\right)^2 \frac{n_{11}}{n_{11}-1}. \tag{4.4c}$$

Let $min(\Delta F)$ be the minimum of {$\Delta F_{11\to 00}$, $\Delta F_{11\to 10}$, $\Delta F_{11\to 01}$}. If $min(\Delta F) < 0$, then move the pixel $p(x, y)$ from its current region to the region which produces minimum ΔF. This is equivalent to changing the sign of $\phi_i(x, y)$ in the pixel's current region ($i = 1, 2$) to the sign of $\phi_i(x, y)$ in the region with the minimum ΔF value. For example, if $min(\Delta F) = \Delta F_{11\to 10}$, then move pixel p(x, y) from region "11" to region "10". This is equivalent to changing $\phi_2(x, y)$ to $-\phi_2(x, y)$, and $\phi_1(x, y)$ remains unchanged, because in these two regions $\phi_1(x, y)$ has the same value. If $min(\Delta F) > 0$, the values of $\phi_i(x, y)$ remain unchanged. Changing the sign of $\phi_i(x, y)$ in the other regions is done in a similar manner.

For the piecewise constant segmentation approach, we implemented both the Chan–Vese active contour approach and the direct energy computation method. Both algorithms give almost the same results for our testing images including synthetic and real medical images.

4.1.3 Remarks on the Chan–Vese Active Contour Models

A. The Edge Detection

One attractive feature associated with the Chan–Vese active contour models is that it can detect contours with or without gradient, for example object with very smooth boundaries or with discontinuous boundaries [9]. For binary images, the 2–phase segmentation model discussed in section 3.1.4 is sufficient to get the correct detection of contours. We show in Fig 4.3 that the edge detection results obtained by using the Chan–Vese active contour model [9] and the results obtained from other edge detection methods for comparisons.

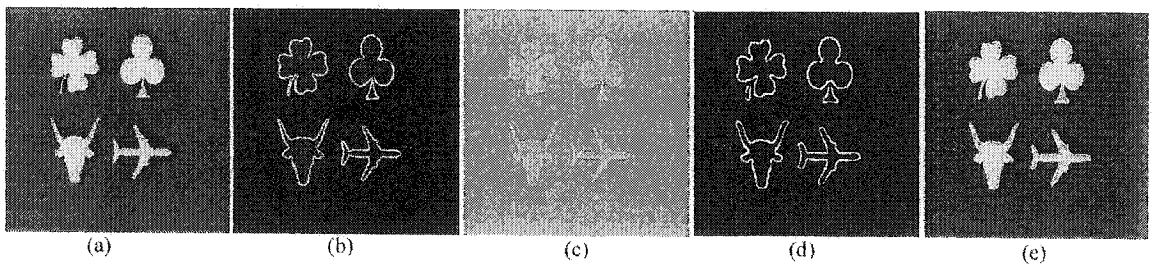


Fig. 4.3. Edge detection using different approaches. (a) Original image. (b) The Sobel gradient method. (c) The Laplacian method. (d) The Marr and Hildreth method. (e) The active contour method.

Fig. 4.4 shows an image with very smooth edges, the difference of the intensity values between the Greek alphabets and the background is only 1 and we cannot even distinguish the objects from the background with human eyes. The Sobel gradient method and the Laplacian method cannot detect these edges, although the Marr and Hildreth method can detect the edges, but the corners are rounded. The active contour model gives the best performance for the detection of very smooth edges. There we can conclude that the Chan–Vese piecewise constant active contour model and the direct energy

computation approach can perform edge detection very well. The direct energy computation approach is more suitable to apply to edge detection because it is very fast, easy to implement, and has good performance.

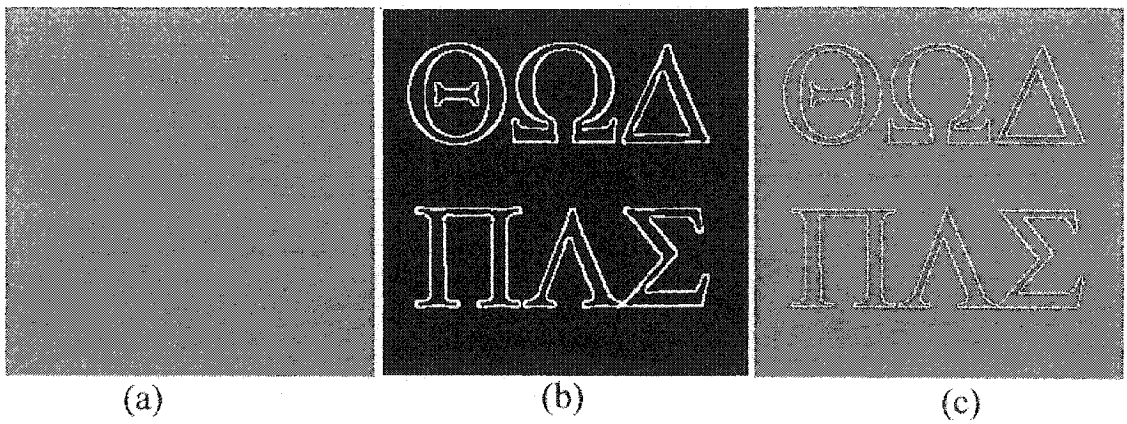


Fig. 4.4. Detect very smooth edges. (a) Original images. (b) The Marr and Hildreth method. (c) The active contour method.

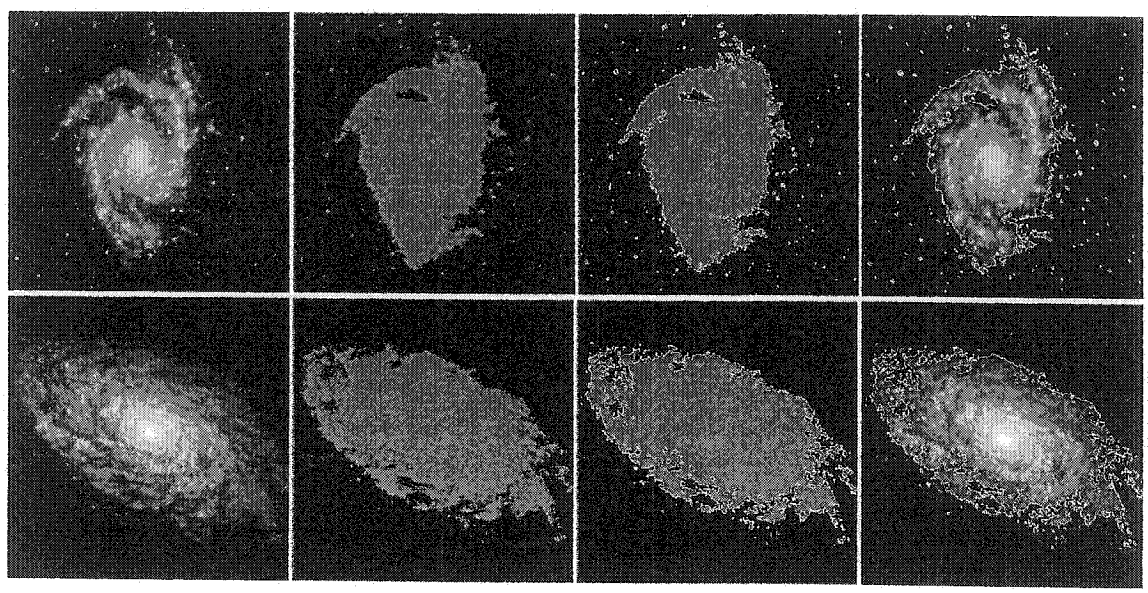


Fig. 4.5. Detection of the contours of galaxies. From left to right, 1st column: Original galaxy images. 2nd column: The piecewise constant segmentation results. 3rd column: The piecewise constant segmentation results with the segmenting curves. 4th column: The piecewise smooth segmentation results with the segmenting curves.

As already discussed by Chan and Vese in [9], *unlike the edge–based active contours models using the gradient as stopping criteria, the Chan–Vese active contour model does not have a stopping edge function. Its stopping term is based on the Mumford–Shah segmentation techniques* [3], *so it can detect contours without edges.* In Fig. 4.5 we use two galaxy images to show how the Chan–Vese active contour model works for images without edges. With the 2–phase approach, we can detect only the outer contours of the galaxies.

B. Problems of the Initial Conditions

It should be noted that the segmentation results obtained by using the Chan–Vese multiphase segmentation method are initial conditions dependent. In Fig. 4.6, we show an experimental example obtained by using the Chan–Vese 4–phase segmentation algorithm with different initial conditions. For the given synthetic image, we get different results of segmentation from different initial curves. For example, when we use four initial circles for the synthetic image (see the far left image in the first row), we can obtain the correct segmentations. But we cannot get the correct segmentations when more initial circles are used. Chan and Vese suggested that multiple initial conditions should be used [4], but we cannot always obtain correct segmentations by using multiple initial conditions (see the far right column of Fig. 4.6). So it is a big challenge to find an efficient way to handle the initial conditions in the Chan–Vese multiphase segmentation model. Another problem with the Chan–Vese segmentation methods is their computational costs. Because of the coupling between the curve evolution equations for ϕ_1 and ϕ_2 (see (4.3a) and (4.3b)), the algorithm is slow, and the final results depend on the choice of the initial curves. The multiple initial curves technique cannot solve the initial curves dependent problem.

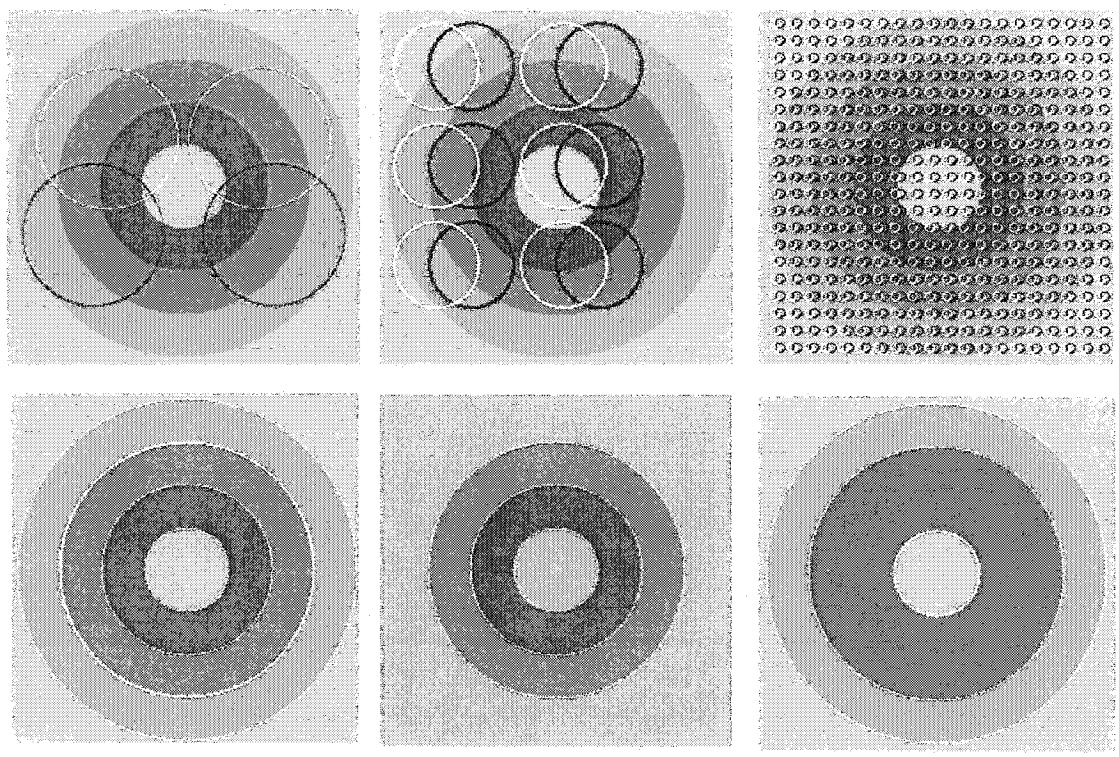


Fig. 4.6. Segmentations in different initial conditions using Chan–Vese multiphase model. The first row shows different initial conditions: white curves for ϕ_1 and black curves for ϕ_2. The second row contains the corresponding segmentations.

In summary, the Chan–Vese active contour model can detect contours with and without edges. One problem of the Chan–Vese multiphase segmentation algorithm is that the computation cost is very high, because it involves solving multiple coupled PDEs for the curve evolution when multiple level set functions are used. For example in the 4–phase segmentation case, when we solve one of the two level set PDEs (4.3a) for instance, we need to compute all the four average values in each region using equations (4.2a)–(4.2d), and the other level set PDE (4.3b) is also involved because it needs the value of ϕ_2. It is the same situation for solving the level set PDE (4.3b). Another problem is the initial conditions are difficult to handle, because the two curves evolve simultaneously when solving the coupled PDEs. The initial condition problem comes

from the minimization problem of the Mumford–Shah energy functional as we have mentioned in section 3.1.3. In general the global minimizer of the Mumford–Shah functional is not unique, we may compute only a local minimum.

In the next section we propose a hierarchical method for multiple phase segmentation in order to overcome the above shortcomings of the Chan–Vese model.

4.2 The Hierarchical Multiphase Segmentation Method

In this section we examine how the parameters associated with the energy inside and outside the contours affect the final segmentation result. Later we present the hierarchical implementation of multiphase piecewise constant segmentation model.

4.2.1 The Effect of the Weight Parameters

It is well known that using one level set function in the Chan–Vese model we can segment an image into different regions with two distinct means. If we consider the parameters associated with the 'fitting energies' in the energy functional, (3.14) and (3.16) can be written as:

$$F_2(c_1,c_2,\phi) = \nu \int_\Omega |\nabla H(\phi)|dxdy + \alpha_1 \int_\Omega |u_0 - c_1|^2 H(\phi)dxdy$$

$$+ \alpha_2 \int_\Omega |u_0 - c_2|^2 (1 - H(\phi))dxdy, \tag{4.5}$$

$$\frac{\partial \phi}{\partial t} = \delta(\phi)\left[\nu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right) - \alpha_1 (u_0 - c_1)^2 + \alpha_2 (u_0 - c_2)^2\right], \tag{4.6}$$

where α_1 and α_2 are parameters associated with the energy inside and outside the segmenting curve respectively. Actually, the energy inside (or outside) the curve is the statistical measurement of an image within the region inside (or outside) the curve. The

parameters α_1 and α_2 behave as weight factors in the statistical measurement. If $\alpha_1 > \alpha_2$, the energy functional inside the curve is more important than that outside the curve, therefore we can get more detailed segmentations inside the curve, and vice versa. Fig. 4.7 shows the experimental results for two given images. The first row in Fig. 4.7 presents (a) a synthetic image with one initial curve, (b) the segmented image by choosing $\alpha_1 = \alpha_2 = 1$, and (c) the segmented image by choosing $\alpha_1 = 1$, $\alpha_2 = 2$. Those results show that when we use the same weight factors in both regions ($\alpha_1 = \alpha_2$), we can only detect two objects out of three. If we want to emphasize the region for $\phi < 0$ by choosing the weight factors $\alpha_2 > \alpha_1$, we can detect all of the three objects with the same initial curve. Fig. 4.7 (d) shows an original MRI knee stir image with one initial curve, (e)–(i) are the segmented results of the MRI image obtained by fixing $\alpha_2 = 1$, and varying α_1. It can be seen that we can get more details of the segmented results by increasing the weight parameter α_1 to a certain limit. When the parameter α_1 becomes too large, some of details of segmentation are lost (see (h) and (i) of Fig. 4.7).

Using different parameters in different regions allows us to obtain more detailed segmentations in specific regions. However, we cannot completely segment an image with multiple distinct means by just increasing certain weight parameters. In order to obtain detailed segmentations of an image and better performance of the algorithm, we propose a hierarchical approach of multiphase segmentation which is faster than the Chan–Vese algorithm. Furthermore, the initial conditions are much easier to handle in the proposed method.

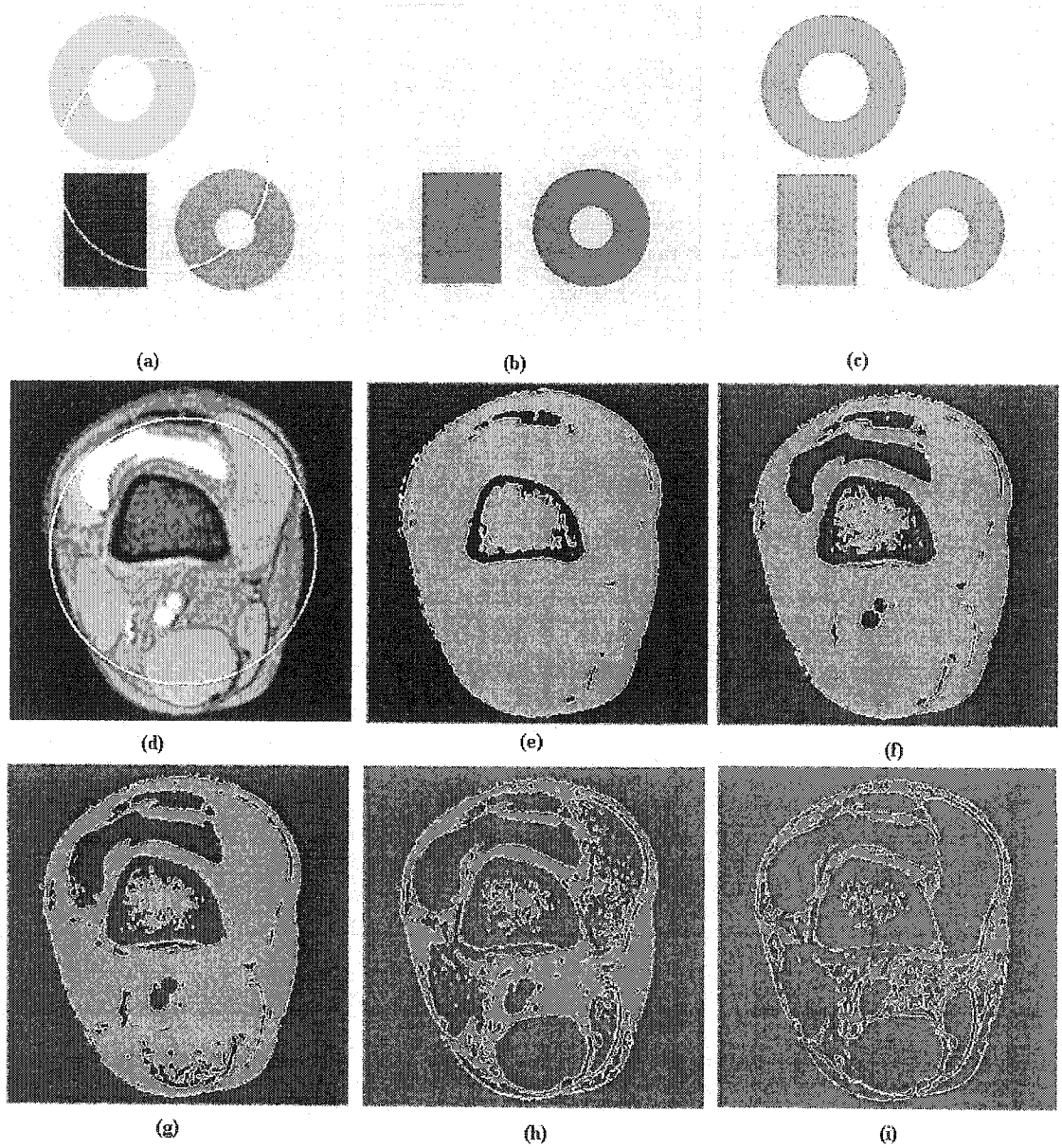


Fig. 4.7. Two–phase segmentation results with different parameters α_1 and α_2. (a) Original synthetic image with an initial curve. (b) $\alpha_1 = 1$, and $\alpha_2 = 1$. (c) $\alpha_1 = 1$, and $\alpha_2 = 2$. (d) Original MRI knee stir image with an initial curve. (e) – (i), $\alpha_2 = 1$. (e) $\alpha_1 = 1$. (f) $\alpha_1 = 3$. (g) $\alpha_1 = 5$ (h) $\alpha_1 = 10$. (i) $\alpha_1 = 20$.

4.2.2 The Hierarchical Method

Using the piecewise smooth representation of the Mumford-Shah model, Tsai et al. have implemented the active contours model in a hierarchical approach [5]. In this approach, they first apply the piecewise smooth segmentation and smoothing algorithm which is described by equations (3.9)–(3.11) to the original image. Then the same algorithm is applied to the particular sub-regions which require additional segmentation. Since only one level set function is used in this approach, and it cannot detect in advance which parts require additional segmentations, therefore this method can only deal with the second segmentation in an ad-hoc manner rather than systematically and automatically as shown by our method to be described in this section. Furthermore, solving the Poison equations (3.9) and (3.10) is very time consuming.

We now present a new hierarchical approach for multiphase piecewise constant segmentation model using multiple level set functions. This approach works in multiple segmentation stages, in the *first* stage, we apply the Chan–Vese piecewise constant segmentation model with one level set function ϕ_1 to a given image. At the end of the *first* segmentation stage, we get *two* resulting sub-regions. Then the second stage starts by applying the same model with another level set function ϕ_2 to each of the sub-regions independently. After the *second* segmentation, we get *four* resulting sub-sub-regions. Next we apply the same model with the third level set function ϕ_3 to each of those sub-sub-regions, and so on, each segmentation applying to the image automatically. In our experience most real images require 2 level set functions. A third level set function may be needed for images with very complicated features.

We illustrate multiphase hierarchical approach through an example of the 4–phase segmentation. Like the Chan–Vese model, there are two evolution curves C_1 and C_2, which are represented by the corresponding level set functions ϕ_1 and ϕ_2. In the *first* segmentation stage, the evolution of curve C_1 is governed by the motion equation of the level set function ϕ_1:

$$\frac{\partial \phi_1}{\partial t} = \delta(\phi_1)[\nu \nabla \cdot \left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right) - \alpha_1 (u_0 - c^I)^2 + \alpha_2 (u_0 - c^{II})^2], \tag{4.7}$$

where c^{I} is the average of u_0 inside curve C_1 ($\phi_1 > 0$) and c^{II} is the average of u_0 outside curve C_1 ($\phi_1 < 0$). At the end of the *first* segmentation stage, we obtain two sub-regions, defined by $\{\phi_1 > 0\}$, and $\{\phi_1 < 0\}$. At the *second* segmentation stage, we apply the curve evolution PDE of C_2 (represented by the level set equation for ϕ_2) to these two sub-regions separately, therefore for sub-region I $\{\phi_1 > 0\}$ we have the curve evolution PDE:

$$\frac{\partial \phi_2}{\partial t} = \delta(\phi_2)[\nu \nabla \cdot \left(\frac{\nabla \phi_2}{|\nabla \phi_2|}\right) - \alpha_1 (u_0 - c_1^I)^2 + \alpha_2 (u_0 - c_2^I)^2], \tag{4.8a}$$

where c_1^I is the average of u_0 in sub-region I and also inside the curve C_2 ($\phi_1 > 0$, $\phi_2 > 0$) and c_2^I is the average of u_0 in sub-region I but outside the curve C_2 ($\phi_1 > 0$, $\phi_2 < 0$).

For sub-region II $\{\phi_1 < 0\}$ the curve evolution PDE is:

$$\frac{\partial \phi_2}{\partial t} = \delta(\phi_2)[\nu \nabla \cdot \left(\frac{\nabla \phi_2}{|\nabla \phi_2|}\right) - \alpha_1 (u_0 - c_1^{II})^2 + \alpha_2 (u_0 - c_2^{II})^2], \tag{4.8b}$$

where c_1^{II} is the average of u_0 in sub-region II and also inside C_2 ($\phi_1 < 0$, $\phi_2 > 0$) and c_2^{II} is the average of u_0 in sub-region II but outside C_2 ($\phi_1 < 0$, $\phi_2 < 0$). After the second segmentation, we obtain the final four segments of a given image. Unlike the Chan–Vese multiphase active contour model [4], [87], in our hierarchical multiphase segmentation

method, instead of applying the level set functions ϕ_1 and ϕ_2 simultaneously to an image, we apply these functions one after another. Thus, the motion equation of ϕ_1 (4.8a) and the motion equations of ϕ_2 (4.8b) are completely decoupled. Each equation alone behaves the same as the curve evolution PDE in the simple two–phase (one level set function) segmentation model, therefore it is fast and its initial condition becomes easy to handle. The *direct energy computation* approach can be implemented in the exact same way.

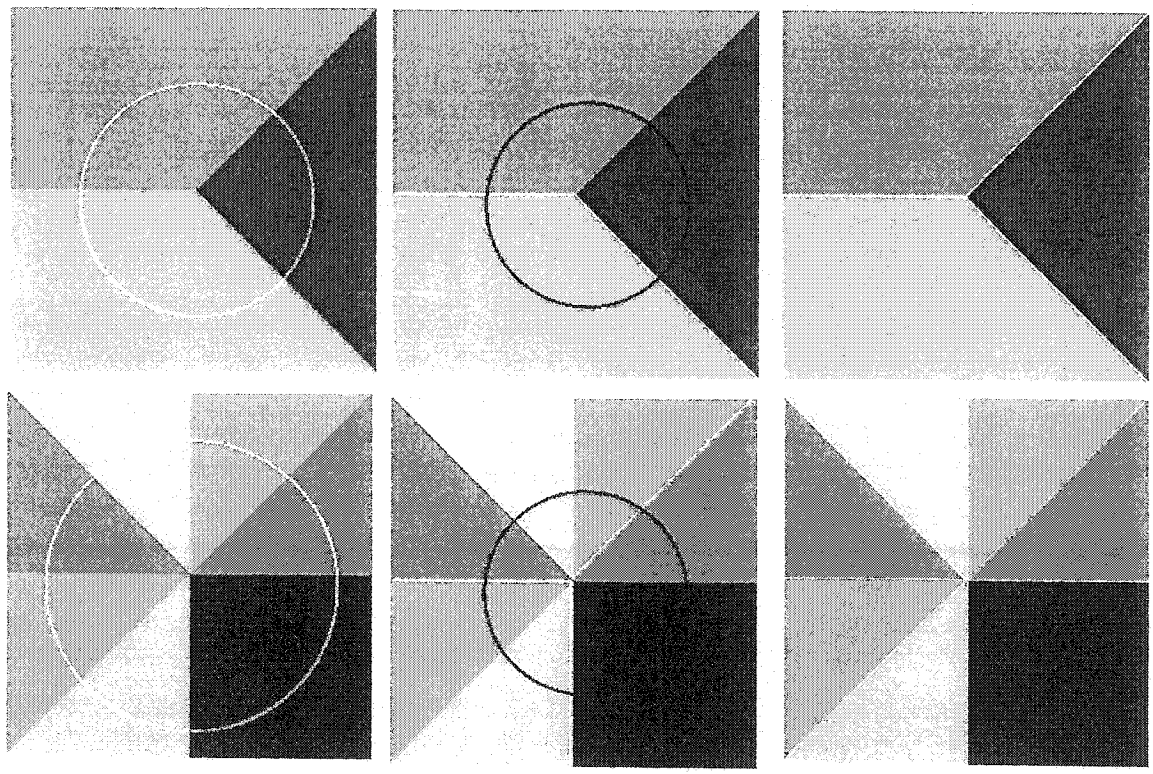


Fig. 4.8. Four–phase segmentation using hierarchical method. 1st column: Original images with the initial curve of ϕ_1. 2nd column: Results after first segmentation (see the white lines) with the initial curve of ϕ_2. 3rd column: Final segmented images with the segmenting lines.

We illustrate how the proposed hierarchical segmentation method works through an example, which is shown in Fig. 4.8. The far left column is the original synthetic images

with the initial curve of ϕ_1. The middle column is the segmented images at the end of the first stage with the initial curve of ϕ_2 superimposed on them. The right column is the final segmentations of the two given images which show that the triple and multiple junctions within the images are detected. From Fig. 4.8 we can see that the first segmentation is like a coarse segmentation of an image, while the second does more detailed segmentation. Actually, this hierarchical method starts with a crude segmentation and refines the segmentation down to the different sub–regions in order to capture finer and finer details in a given image.

The initial condition is easy to handle in the hierarchical method, because curve evolution PDEs are decoupled. At each segmentation stage, only one curve evolution equation represented by a single level set function is involved. Therefore, just one level set function needs to be initialized. In one level set evolution case, we can get sufficient result by using one single initial curve [9], [85]. In our implementations, we use a single initial curve for the level set function ϕ_1 at the first segmentation stage to get a crude segmentation. Although we also use a single curve to initialize ϕ_2 for the second segmentation stage, the actual initial condition of ϕ_2 is multiple curves because of the presence of the final stage of ϕ_1. The edges obtained by the first segmentation (white curves in the middle column of Fig. 4.8) together with the initial curve of ϕ_2 (the black curve in the middle column of Fig. 4.8) make the actual initial condition of ϕ_2, which is optimal, multiple curves.

In this hierarchical approach, edges obtained from the first stage together with the initial curve of the second level set function automatically construct the real initial condition of the curve evolution equation at the second stage. Edges obtained at the end

of second stage together with the initial curve of the third level set function construct the actual initial condition of the curve evolution equation at the third stage, and so on. We can get sufficient segmentations by using single initial curve for each individual level set function, as long as the initial curve contains the regions (objects) we want to segment.

Like the Chan–Vese active contour model, the proposed hierarchical approach is based on the Mumford–Shah energy functional, its stopping criteria comes from the Mumford–Shah segmentation technique, it can detect contours within an image with or without edges. Using the same galaxy images as in Fig. 4.5, the 4–phase hierarchical method produces more detailed segmentations than the 2–phase active contour model and the contour within these images are detected. The final results are shown in Fig. 4.9.

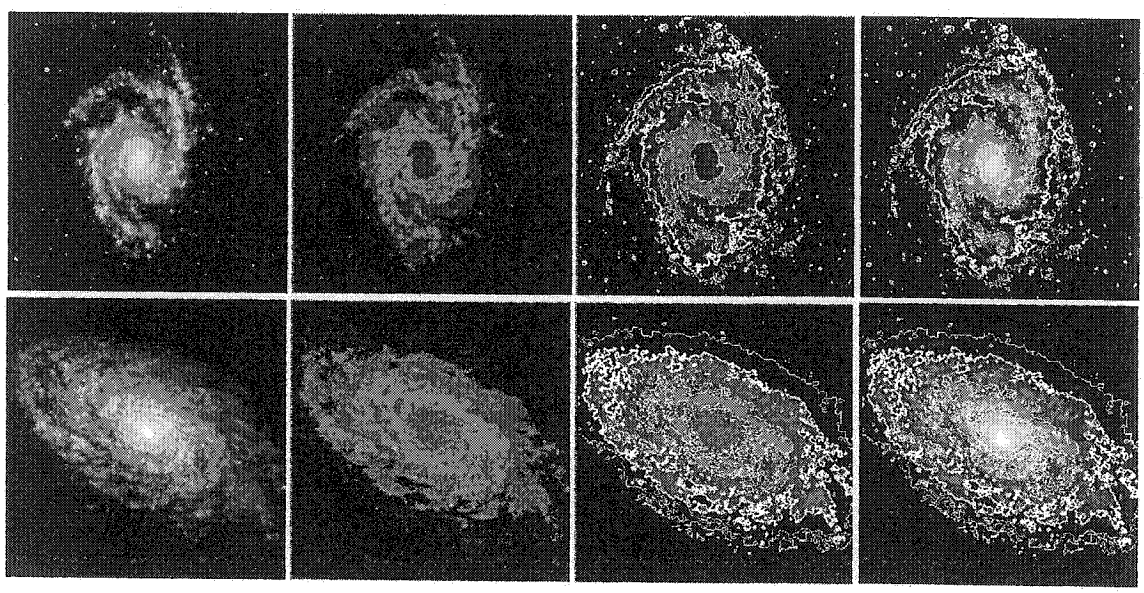
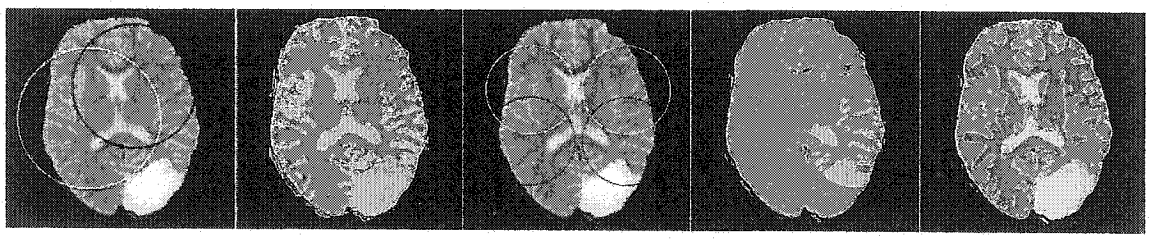


Fig. 4.9. Detection of the contours of galaxy images using the 4–phase hierarchical method. From left to right, 1st column: Original galaxy images. 2nd column: The piecewise constant segmentation results. 3rd column: The piecewise constant segmentation results with the segmenting curves. 4th column: The piecewise smooth segmentation results with the segmenting curves.

In the next experimental example we use an MRI medical image to demonstrate the proposed hierarchical segmentation method is more efficient than the Chan–Vese multiphase segmentation algorithm, the results are given in Fig. 4.10. This figure shows that the hierarchical method works for ν=1, while the Chan–Vese algorithm does not work properly for ν =1, but it tendencies to slow-down the algorithm using big value of ν. The effects of parameter ν need more future investigations. We also compared the machine time for those two algorithms by running our programs on Pentium IV 2.40GHz. The result shows that our proposed segmentation method is faster than the Chan–Vese algorithm.



(*a*) (*b*) (*c*) (*d*) (*e*)

Fig. 4.10. Segmentation of an MRI image using the hierarchical and the Chan–Vese algorithms. (*a*) Initial conditions for the hierarchical method. (*b*) Segmented results using the hierarchical method, ν=1, time=0:07min. (*c*) Initial conditions for the Chan–Vese method. (*d*) Segmented results using the Chan–Vese method, ν=1, time=0:17min. (*e*) Segmented results using the Chan–Vese method, ν=300, time=1:07min. Image size: 256×256.

As we already mentioned in section 3.1.4, the piecewise constant segmentation method represents each segment of an image by a constant in each region. The most general Mumford-Shah segmentation method is the piecewise smooth representation of an image, which has been addressed in [4], [5], [87]. In order to get better performance for denoising and segmentation of an image, we propose a new two–step segmentation and smoothing method in the following chapter.

Chapter 5

The Two–step Segmentation and Selective Smoothing Method

This chapter presents the segmentation prior to smoothing method for image segmentation, denoising, and edge enhancement. This method combines the Mumford–Shah energy functional based hierarchical segmentation algorithm with the PDE–based image smoothing and denoising method. This algorithm performs segmentation and denoising in a unified way and the smoothing process is very similar to the anisotropic diffusion process. This method gives better denoising results for very noisy image than that given by using the Chan–Vese piecewise smooth active contour model.

5.1 The Mumford–Shah Active Contour Model

The active contour model based on the Mumford–Shah functional actually performs image segmentation and smoothing simultaneously. As we have discussed in section 3.1.4, the simultaneous image segmentation and smoothing algorithm is obtained by minimizing the *original* Mumford–Shah functional [4], [5] with respect to the image values $u_i(x, y)$ in different regions and the active contours C_i. For 2–phase segmentation in which the active contour is represented by *one* level set function ϕ, this algorithm is

presented by the *Euler–Lagrange* equations (3.9)–(3.11). The evolution of the active contour is controlled by (3.11) whereas (3.9) and (3.10) smooth the image data in different regions. The smoothing process is the linear diffusion within each homogenous region which is obtained from the segmentation of an image, and the edge is preserved because smoothing is not across the edges.

5.1.1 Multiphase Piecewise Smooth Model

The piecewise smooth algorithm can also be extended to multiphase case by using more than one level set function [4], [87]. Like the multiphase piecewise constant approach we have discussed in chapter 4, using two level set functions ϕ_1 and ϕ_2 we can identify four different regions (see Fig. 4.2(b)). Let us introduce four functions $u_{11}(x, y)$, $u_{10}(x, y)$, $u_{01}(x, y)$, and $u_{00}(x, y)$ to represent the value of the image function $u(x, y)$ in each of the four regions, as follows:

$$u(x,y) = \begin{cases} u_{11}(x,y), & \{\phi_1(x,y) > 0, \text{and } \phi_2(x,y) > 0\} \\ u_{10}(x,y), & \{\phi_1(x,y) > 0, \text{and } \phi_2(x,y) < 0\} \\ u_{01}(x,y), & \{\phi_1(x,y) < 0, \text{and } \phi_2(x,y) > 0\} \\ u_{00}(x,y), & \{\phi_1(x,y) < 0, \text{and } \phi_2(x,y) < 0\} \end{cases}$$

Using the Heaviside functions $H(\phi_1)$ and $H(\phi_2)$, we can represent the image function $u(x, y)$ in terms of the four functions u_{11}, u_{10}, u_{01}, and u_{00} as,

$$u = u_{11}H(\phi_1)H(\phi_2) + u_{10}H(\phi_1)(1-H(\phi_2)) + u_{01}(1-H(\phi_1))H(\phi_2) + u_{00}(1-H(\phi_1))(1-H(\phi_2)).$$

Based on these notations, we can obtain the *level set formulation* of the Mumford–Shah *original* energy functional (3.7) for the 4–phase representation of an image domain as follows,

$$F_4(u,\Phi) = \int_\Omega |u_0 - u_{11}|^2 H(\phi_1)H(\phi_2)\, dxdy + \mu \int_\Omega |\nabla u_{11}|^2 H(\phi_1)H(\phi_2)\, dxdy$$

$$+\int_{\Omega}|u_0-u_{10}|^2\,H(\phi_1)(1-H(\phi_2))\,dxdy+\mu\int_{\Omega}|\nabla u_{10}|^2\,H(\phi_1)(1-H(\phi_2))\,dxdy$$

$$+\int_{\Omega}|u_0-u_{01}|^2\,(1-H(\phi_1))H(\phi_2)\,dxdy+\mu\int_{\Omega}|\nabla u_{01}|^2\,(1-H(\phi_1))H(\phi_2)\,dxdy$$

$$+\int_{\Omega}|u_0-u_{00}|^2\,(1-H(\phi_1))(1-H(\phi_2))\,dxdy$$

$$+\mu\int_{\Omega}|\nabla u_{00}|^2\,(1-H(\phi_1))(1-H(\phi_2))\,dxdy$$

$$+\nu\int_{\Omega}|\nabla H(\phi_1)|\,dxdy+\nu\int_{\Omega}|\nabla H(\phi_2)|\,dxdy\,. \qquad (5.1)$$

The last two terms are the approximations of the length term of the curves C_1 and C_2. The *active contour* (dynamical scheme) of this model can be represented by introducing the time–dependent level set functions $\phi_1(x, y, t)$ and $\phi_2(x, y, t)$ which correspond to the active contours $C_1(x, y, t)$ and $C_2(x, y, t)$ respectively. Fixing the time t and minimizing the energy functional (5.1) with respect to the functions u_{11}, u_{10}, u_{01}, and u_{00}, we get the Euler–Lagrange equations for u_{11}, u_{10}, u_{01}, and u_{00} as follows,

$$u_{11}-u_0=\mu\nabla^2u_{11} \qquad \text{in } \{\phi1>0 \text{ and } \phi2>0\},$$

$$\frac{\partial u_{11}}{\partial\vec{n}}=0 \qquad \text{on } \{\phi_1=0 \text{ and } \phi_2\geq 0\} \text{ and } \{\phi_1\geq 0 \text{ and } \phi_2=0\}; \qquad (5.2a)$$

$$u_{10}-u_0=\mu\nabla^2u_{10} \quad \text{in } \{\phi_1>0 \text{ and } \phi_2<0\},$$

$$\frac{\partial u_{10}}{\partial\vec{n}}=0 \qquad \text{on } \{\phi_1=0 \text{ and } \phi_2\leq 0\} \text{ and } \{\phi_1\geq 0 \text{ and } \phi_2=0\}; \qquad (5.2b)$$

$$u_{01}-u_0=\mu\nabla^2u_{01} \quad \text{in } \{\phi_1<0 \text{ and } \phi_2>0\},$$

$$\frac{\partial u_{01}}{\partial\vec{n}}=0 \qquad \text{on } \{\phi_1=0 \text{ and } \phi_2\geq 0\} \text{ and } \{\phi_1\leq 0 \text{ and } \phi_2=0\}; \qquad (5.2c)$$

$$u_{00}-u_0=\mu\nabla^2u_{00} \quad \text{in } \{\phi_1<0 \text{ and } \phi_2<0\},$$

$$\frac{\partial u_{00}}{\partial \vec{n}} = 0 \qquad \text{on } \{\phi_1 = 0 \text{ and } \phi_2 \le 0\} \text{ and } \{\phi_1 \le 0 \text{ and } \phi_2 = 0\}. \tag{5.2d}$$

The curve evolution PDEs for $\phi_1(x, y, t)$ and $\phi_2(x, y, t)$ are obtained by minimizing the energy functional (5.1) with respect to ϕ_1 and ϕ_2 respectively as follows,

$$\begin{aligned}\frac{\partial \phi_1}{\partial t} = \delta(\phi_1)\Big[& \nu \nabla \cdot \left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right) - |u_{11} - u_0|^2 H(\phi_2) - \mu |\nabla u_{11}|^2 H(\phi_2) \\ & - |u_{10} - u_0|^2 (1 - H(\phi_2)) - \mu |\nabla u_{10}|^2 (1 - H(\phi_2)) \\ & + |u_{01} - u_0|^2 H(\phi_2) + \mu |\nabla u_{01}|^2 H(\phi_2) \\ & + |u_{00} - u_0|^2 (1 - H(\phi_2)) + \mu |\nabla u_{00}|^2 (1 - H(\phi_2))\Big], \end{aligned} \tag{5.3a}$$

$$\begin{aligned}\frac{\partial \phi_2}{\partial t} = \delta(\phi_2)\Big[& \nu \nabla \cdot \left(\frac{\nabla \phi_2}{|\nabla \phi_2|}\right) - |u_{11} - u_0|^2 H(\phi_1) - \mu |\nabla u_{11}|^2 H(\phi_1) \\ & + |u_{10} - u_0|^2 H(\phi_1) + \mu |\nabla u_{10}|^2 H(\phi_1) \\ & - |u_{01} - u_0|^2 (1 - H(\phi_1)) - \mu |\nabla u_{01}|^2 (1 - H(\phi_1)) \\ & + |u_{00} - u_0|^2 (1 - H(\phi_1)) + \mu |\nabla u_{00}|^2 (1 - H(\phi_1))\Big]. \end{aligned} \tag{5.3b}$$

The values of the functions u_{11}, u_{10}, u_{01}, and u_{00}, are obtained by solving the PDEs (5.2a)–(5.2d) for some given curves C_1 and C_2, These PDEs are damped *Poisson* equations that can smooth the image data. The evolving of the active contours C_1 and C_2 is governed by the PDEs (5.3a) and (5.3b) respectively. Like the curve evolution PDEs (4.3a) and (4.3b) in the piecewise constant approach, these two PDEs are highly coupled that makes the algorithm slow and the numerical results depend on the choice of initial curves.

5.1.2 Comments on the Piecewise Smooth Model

Although some advantages can be achieved for image segmentation and denoising in this piecewise smooth approach, such as simultaneous segmentation and smoothing of noisy images, detection of triple junctions by using multiple level set functions, and smoothing the images with complex features [4], [5]. However, in this piecewise smooth approach, there are some disadvantages we would like to address here:

i) The capability of denoising is limited because of the damped Poisson equation used for denoising. Furthermore, because the Poisson equations for u_i have the same form and parameters, the same amount of noise will be removed in the homogeneous regions divided by the active contours C_i. In some applications this restriction may limit the flexibility of the system.

ii) Since this algorithm performs segmentation and denoising simultaneously, when the image is very noisy, and noise may destroy some parts of the edges, one may not be able to obtain correct segmentations and smoothing of the image.

iii) Computation cost is another problem in this algorithm. For 2–phase segmentation that uses only *one* level set function, there are *three* PDEs to be solved ((3.9)–(3.11)) and solving the curve evolution equation (3.11) involves the other two equations (3.9) and (3.10) in each iteration step. However, only the solutions of (3.9) and (3.10) at the final step are the values of the smoothed image. For the 4–phase segmentation case, since we need to use *two* level set functions, there are totally *six* PDEs ((5.2a) –(5.2d) and (5.3a)–(5.3b)) needed to be solved. Solving each curve evolution equation (5.3a) for instance (it is same for solving (5.3b)) involves all the other *five* equations, because it needs the values of the functions

u_{11}, u_{10}, u_{01}, and u_{00} from (5.2a)–(5.2b) and the value of ϕ_2 from (5.3b). So this algorithm is not efficient for numerical implementations. Although many speed-up methods have been proposed in [5], it may not work well in practice when image size is large and the noise ratio is high.

iv) Like the Chan–Vese multiphase piecewise constant segmentation approach, if we apply this piecewise smooth method to multiphase segmentation, because of the coupling of the level set PDEs (5.3a) and (5.3b), handling the initial condition is also an important problem.

Based on the hierarchical approach for multiphase image segmentation we have proposed in chapter 4 and the diffusion equation for image smoothing we have discussed in chapter 3, we propose a new two-step algorithm that performs image segmentation and denoising all together but in sequence. At the first step we use the proposed hierarchical segmentation approach to find the regions within a given image. Then we apply the diffusion filter to each homogeneous region independently but not across the boundaries of such regions.

5.2 The Segmentation Prior to Denoising Method

5.2.1 The Algorithm

The basic idea of our segmentation prior to denoising algorithm is to obtain different sub-regions within a given image by using the piecewise constant segmentation method first, then select each sub-region of the original noisy image as the initial condition (input) of the diffusion equation for the corresponding sub-region. The final reconstruction of an

image is obtained in terms of the combination of the smoothing results in all such sub–regions. Therefore we can smooth each sub-region separately but not across the edges between the sub-regions. This algorithm works in the following steps:

1) Convolve the original noisy image $u(x, y)$ with a Gaussian filter (2.10) of standard deviation σ, to obtain the regularized image $u_\sigma(x, y)$, which makes the edge detection insensitive to noise at scales smaller than σ.

2) Apply the piecewise constant segmentation method to the regularized image $u_\sigma(x, y)$, and partition the image into different regions.

3) After segmentation of the regularized image $u_\sigma(x, y)$ in step *2)*, for example in 2–phase segmentation case, $u_\sigma(x, y)$ is segmented into two regions, one for $\{\phi > 0\}$, and another for $\{\phi < 0\}$. The result of segmentation in step *2)* (i.e. values of ϕ) is now applied to the original image $u(x, y)$, so that both $u_\sigma(x, y)$ and $u(x, y)$ have the same segmentation. Let $u^{\mathrm{I}}(x, y)$ be the original image in region for $\{\phi > 0\}$, and $u^{\mathrm{II}}(x, y)$ be the original image in region for $\{\phi < 0\}$. We then apply the diffusion filter to the different regions of $u(x, y)$ independently. In order to solve the diffusion equations in different regions properly, we need to extend $u^{\mathrm{I}}(x, y)$ to the region $\{\phi < 0\}$ and $u^{\mathrm{II}}(x, y)$ to the region $\{\phi > 0\}$. For instance, to extend $u^{\mathrm{I}}(x, y)$ to the region $\{\phi < 0\}$, we can use the average constant approximation of $u(x, y)$ in region $\{\phi < 0\}$. Other extension methods can be found in [4]. Attention must be paid to the boundaries between the regions. We can use the Neumann boundary conditions $\partial u^{I}(x,y)/\partial \vec{n}=0$ or $\partial u^{II}(x,y)/\partial \vec{n}=0$ ($\vec{n}$ is the normal of the curve C) when we extend $u^{\mathrm{I}}(x, y)$ or $u^{\mathrm{II}}(x, y)$ across the edges between regions. Therefore the diffusion does not cross the boundaries of different regions.

This method is fast since detecting the boundary of the regions only requires updating the average values inside and outside the active contours. This method is more flexible since we can choose different diffusion parameters (or different smoothing methods) for different sub-regions depending on the applications. It can process very noisy images without difficulties. This may be useful in applications such as medical image segmentation and smoothing. The previous piecewise smooth algorithm [4], [5], [87] gives poor results for very noisy images.

We use the *signal-to-noise ratio* (*SNR*) to estimate the quality (or the amount of noise) of the image $u(x, y)$ with respect to a reference image (usually the 'clear' image) $u_0(x, y)$. It is defined by [88]:

$$SNR = 10\log_{10}\left(\frac{\sigma^2(u)}{\sigma^2(u_0 - u)}\right), \tag{5.4}$$

where σ is the variance. For the size of $m \times n$ images, $\sigma^2(u) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}|u(i,j)|^2$, and $\sigma^2(u_0 - u) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}|u_0(i,j) - u(i,j)|^2$. SNR is usually expressed in dB (*decibel*).

5.2.2 Regularizations

An important improvement of the PDE–based filtering theory has been introduced by Malik and Perona [15], they replace the heat equation (3.18) by a nonlinear partial differential equation of the porous medium type:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(g(|\nabla u|)\nabla u\right), \qquad u(x, y, t = 0) = u_0(x, y), \tag{5.5}$$

where $g(|\nabla u|)$ is a scalar smooth function with the properties we addressed in section 3.2.1. As already pointed out by Catte et al. in [16], the Malik and Perona model has

several serious practical and theoretical problems. In practice, although this model performs nonlinear diffusion it cannot eliminate noise at the edges. Equation (5.5) itself also poses difficulties because there is no theory to address existence and uniqueness of the solution for the functions of the type $g(|\nabla u|) = \exp\left(-|\nabla u|^2/\lambda^2\right)$, or $g(|\nabla u|) = 1/\left(1+|\nabla u|^2/\lambda^2\right)$, where λ is a constant. Two regularization approaches have been proposed to overcome these difficulties, one approach was proposed by Catte *et al.* [16] through the regularization of the function $g(|\nabla u|)$ in equation (5.5). Thus the modification of the Malik and Perona model is only to replace the gradient $|\nabla u|$ by its regularized version $|\nabla(K_\sigma * u)|$, where K_σ is a Gaussian filter. The other approach proposed by Weickert [21], [25] uses regularized nonlinear anisotropic diffusion filters which use a diffusion tensor instead of a scalar diffusivity function. Regularization is widely used in PDE–based filtering including anisotropic diffusion [24]–[25]. This is the motivation for performing regularization by convolving a Gaussian filter with the original noisy image in the first step of our proposed denoising algorithm.

Under the framework of the proposed algorithm, we can get the proper sub–regions of a noisy image by applying the hierarchical segmentation algorithm to the regularized image. After that we apply the appropriate filters to the resulting sub–regions to obtain the smooth version of the given image. In general, we can apply any kind of smoothing algorithm to the different homogeneous sub–regions independently. In this thesis, we use the linear diffusion filter defined by (3.18) with a scalar constant diffusivity D under the Neumann boundary conditions. The numerical implementation techniques for solving this type of diffusion PDE can be found in the appendix.

In the first example of this chapter, we demonstrated that performing the regularization of the noisy image before segmentation and smoothing can get better results than without regularization. The regularization process is important especially for very noisy images. Using a synthetic image and a MRI brain image as examples, we show in Fig. 5.1 different results obtained by employing our two–step segmentation and smoothing method with and without regularization of the noisy images. We use the signal-to-noise ratio *SNR* to measure the amount of noise in the noisy images, the higher the signal-to-noise ratio *SNR* the lower the amount of noise by definition (5.4). The images with additive Gaussian noise are shown in the 2nd column of Fig 5.1, for the binary synthetic image (first row) $SNR = 6.89$, and $SNR = 9.20$ for the medical image (second row). Fig. 5.1 shows that the denoising results using regularized images (fourth column) are better than that using the noisy images without regularization (third column).

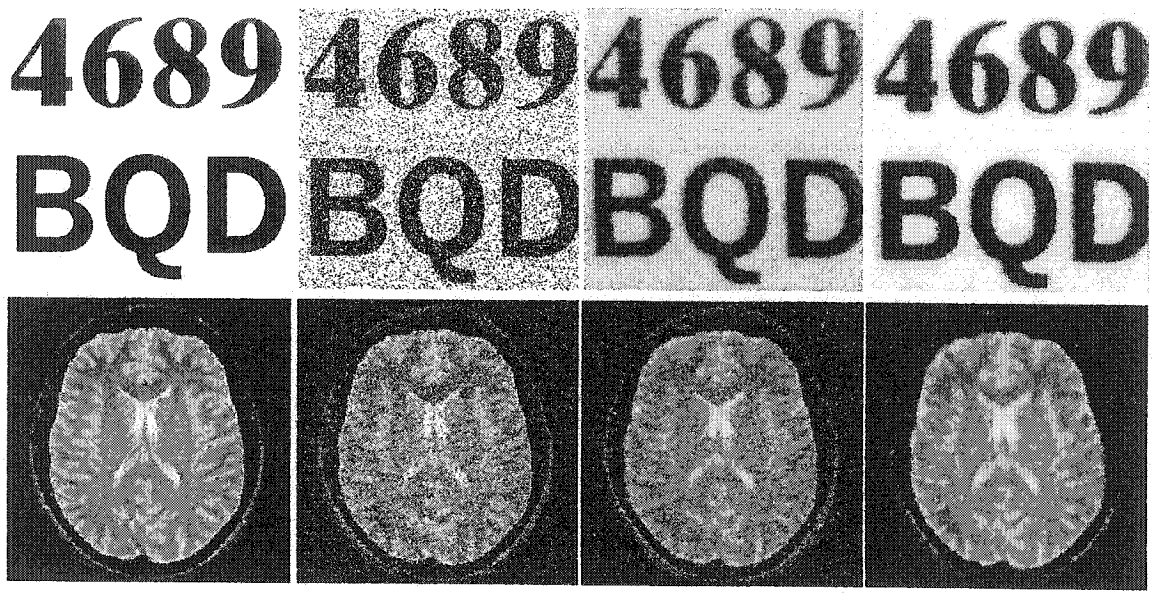


Fig. 5.1. The denoising results with and without regularization. From left to right, 1st column: Original images. 2nd column: Images with Gaussian noise. 3rd column: Smoothing results without regularization. 4th column: Smoothing results with the regularized images.

5.2.3 Comparison with the Chan and Vese Piecewise Smooth Model

The Chan–Vese piecewise smooth active contour model [4], [87] and the similar approach developed by Tsai, Yezzi and Willsky [5] perform image smoothing and segmentation simultaneously, the smoothing process governed by the damped Poisson equations (for 2–phase (3.9) and (3.10), whereas (5.2a)–(5.2d) for 4–phase case). Besides the computing costs and the initialization problem, this model cannot give good denoising results for very noisy images. Our hierarchical segmentation approach presented in chapter 4 can overcome both the computing costs and the initialization problem. Furthermore, since we use the diffusion PDEs instead of the damped Poisson equations to handle the denoising problem, our method can handle very noisy images.

In Fig. 5.2, for comparison, we show the segmentation and denoising results obtained by using the piecewise smooth active contour model developed by Chan and Vese [4], and our proposed method. For an image with small amount of noise (a) (SNR = 16.29 dB), the smoothing image (b) obtained by using the piecewise smooth algorithm [4], [5] is acceptable. If the image is very noisy (d) (SNR = 2.69 dB), the smoothed image (e) obtained by using the Chan–Vese piecewise smooth approach is not as good as desired. Fig. 5.2(h) is the results of the linear isotropic diffusion filtering. The edges of the objects within the image get very blurred when noise is removed. The smoothing results (c and f) obtained by using our proposed two–step segmentation and smoothing method are better than that of the Chan and Vese piecewise smooth model. Unlike the isotropic diffusion that smoothes the whole image in the same way and the edges between regions within the image are also affected. Our algorithm first segments a given image into sub–regions, then the diffusion equation is applied within each sub-region but not across the edges,

therefore the edges are preserved. This is very similar to the smoothing process in anisotropic diffusion [17], [21], [25], and our smoothing result (f) is similar to the smoothing result (i) using Weickert's anisotropic diffusion algorithm [21], [25].

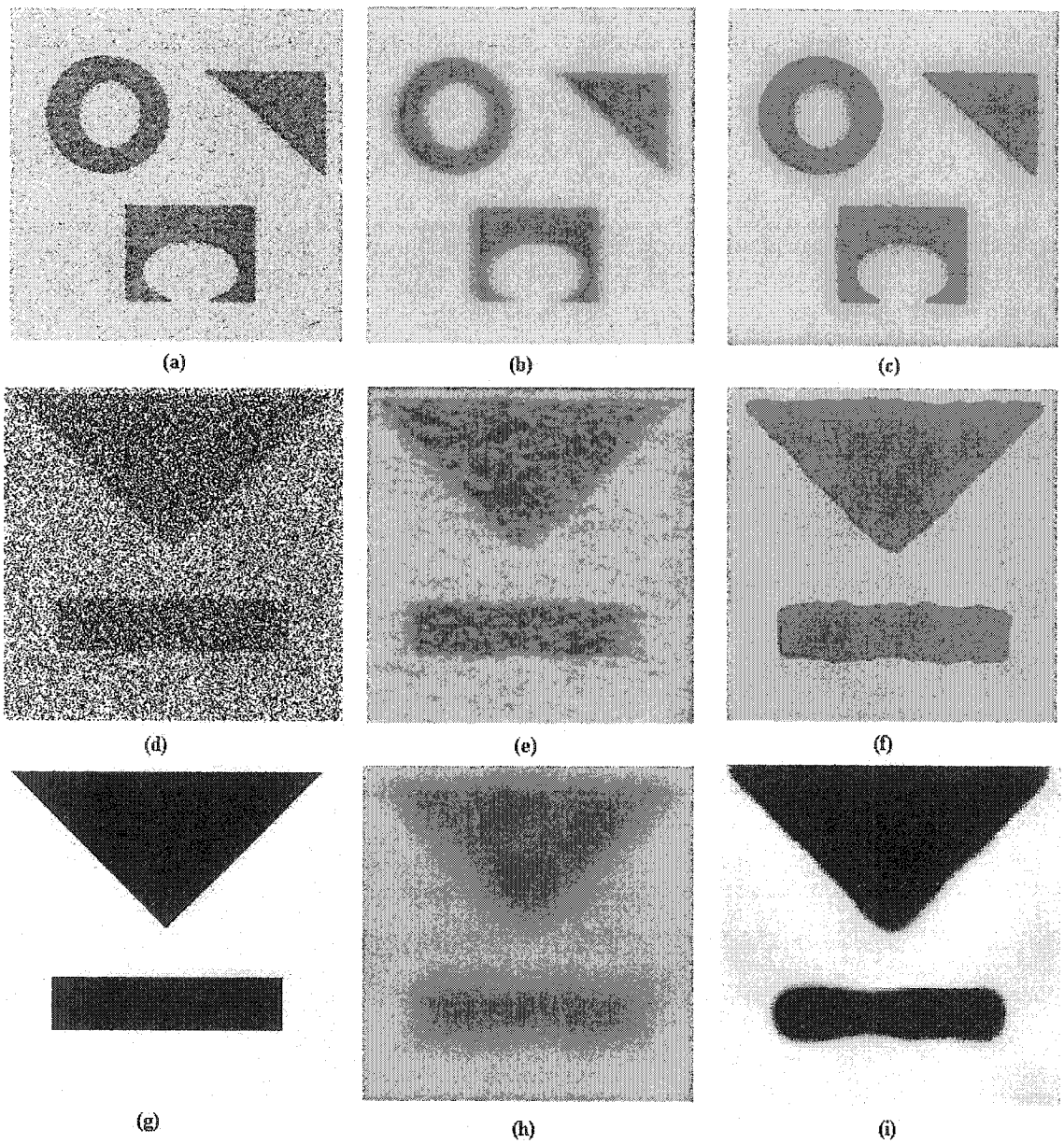


Fig. 5.2. Segmentation and smoothing of noisy synthetic images using different approaches. (a) and (d): Original noisy images. (b) and (e): Reconstructions by Chan–Vese piecewise smooth approach. (c) and (f) Reconstructions by our proposed approach. (g) Original clean image. (h) Reconstruction of image (d) by linear diffusion filtering. (i) Smoothed image obtained by Weickert's anisotropic diffusion [21].

5.2.4 Edge Enhancement

As we have already mentioned, the proposed segmentation and smoothing method which adapts itself in an anisotropic way to the evolving image, is well suited for smoothing noise while simultaneously preserving important features such as edges. This characteristic may be important in medical image processing.

In recent years, many new imaging techniques such as Magnetic Resonance Imaging (MRI), x–ray, x–ray tomography, ultrasound imaging, Computed Tomography (CT), etc. have been widely used in modern biomedical research and practice [77], [83], [84], [89]–[91]. In medical image processing, segmentation is an objective measurement of an anatomical structure location represented by some set of contours, boundaries, or shapes. These boundaries between anatomical structures can be very useful for diagnosing diseases and tumor detection. Our segmentation prior to smoothing method can perform segmentation, denoising, and edge enhancement in a unified way which may meet the requirements of medical image processing. Furthermore, this level set based approach can also be extended to 3D image processing [92], [93] without difficulties.

In Fig. 5.3, we show an example where a smoothing and edge enhancing process has been applied to a MRI brain image. The result obtained by using our proposed method is shown in (d). Compared with the result using the edge enhancing anisotropic diffusion filter developed by Weickert [24] which is shown in (b), the proposed method gives better edge enhancing results. We attribute this to the fact that the edge detection is not based on the gradient of the image in our method. Fig. 5.3(c) shows the segmenting curves superimposed on the resulting image.

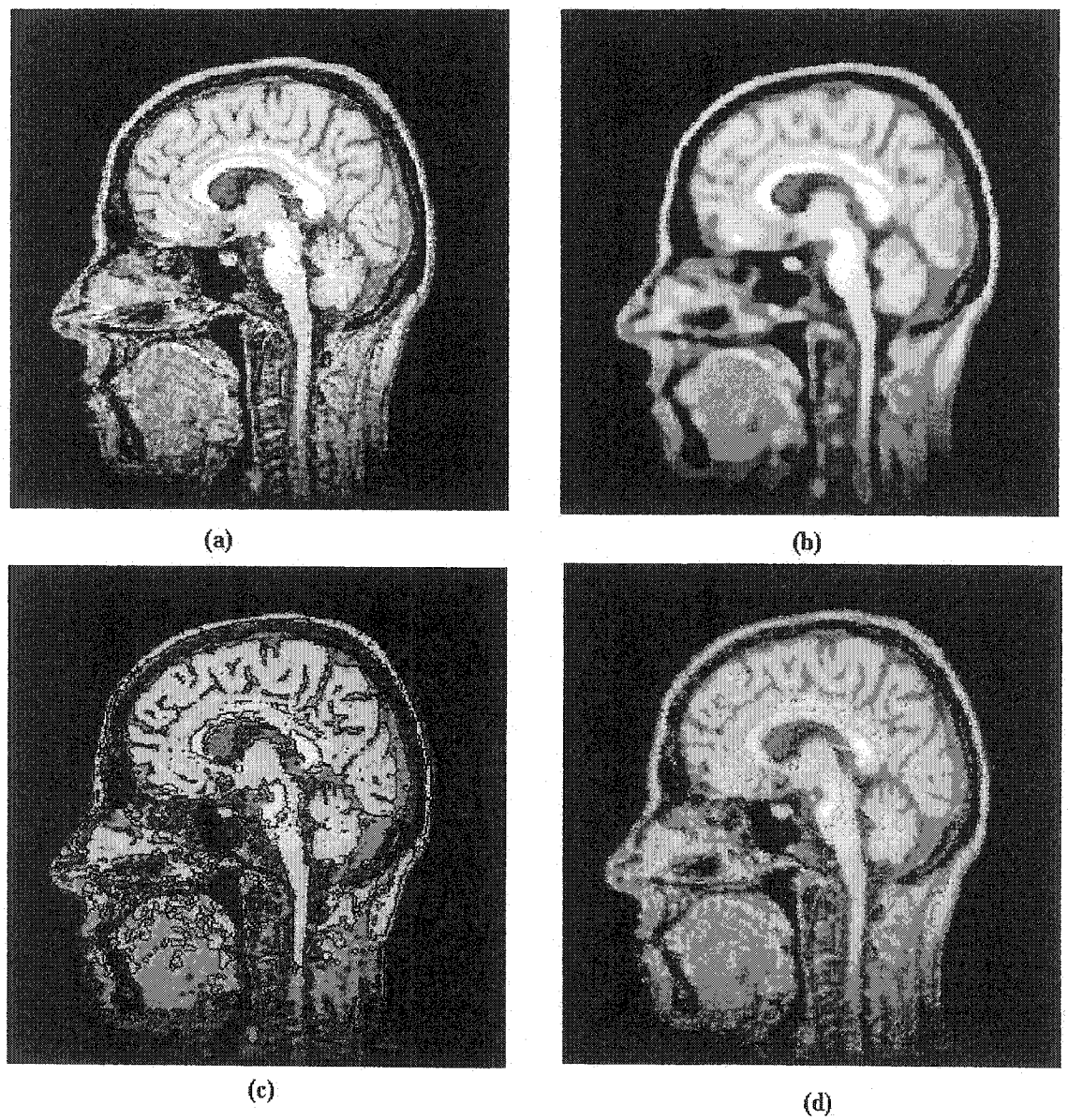


Fig. 5.3. Smooth and edge-enhancement of a medical image. (a) Slice of an MRI image. (b) After anisotropic edge-enhancing diffusion filtering by Weickert [24]. (c) After segmentation and smoothing, segmenting curves (black curves) are superimposed on the image. (d) Final result after segmentation and smoothing of the image.

Chapter 6

Experimental Results

We present in this chapter the segmentation results of the images without noise using our hierarchical segmentation method, followed by the segmentation and smoothing results of the proposed two–step segmentation and smoothing algorithm for noisy images. The proposed method has high performance for image segmentation, denoising, and edge enhancement.

6.1 Segmentation Results

Since the energy functional which is minimized in the Mumford–Shah model is not convex; the segmentation algorithm may not converge to a global minimum for some given initial conditions. As we have shown in Fig. 4.6, it is difficult to handle the initial conditions in the Chan–Vese multiphase active contour model [4]. In the following examples we show that it is easy to handle the initial conditions in our hierarchical segmentation method because we decouple the curve evolution PDEs (4.3). In Fig. 6.1 we present the segmentation results for a synthetic image and a real *x–ray* hand image with different initial curves. The synthetic image is the same image used in Fig.4.6. The detected edges are superimposed on top of the resulting segmented images. For a given image, all those different initial curves give the same segmentation results. Our observation is that, as long as the initial curve contains (or partially contains) the regions

we want to detect, we can obtain the same segmentation results regardless of the positions and sizes of the initial curves. Therefore the problem of choosing the initial curves becomes easy to handle. We have also compared the computational time spent in the hierarchical and the Chan–Vese algorithms, and the machine we used is Pentium IV 2.40GHz.

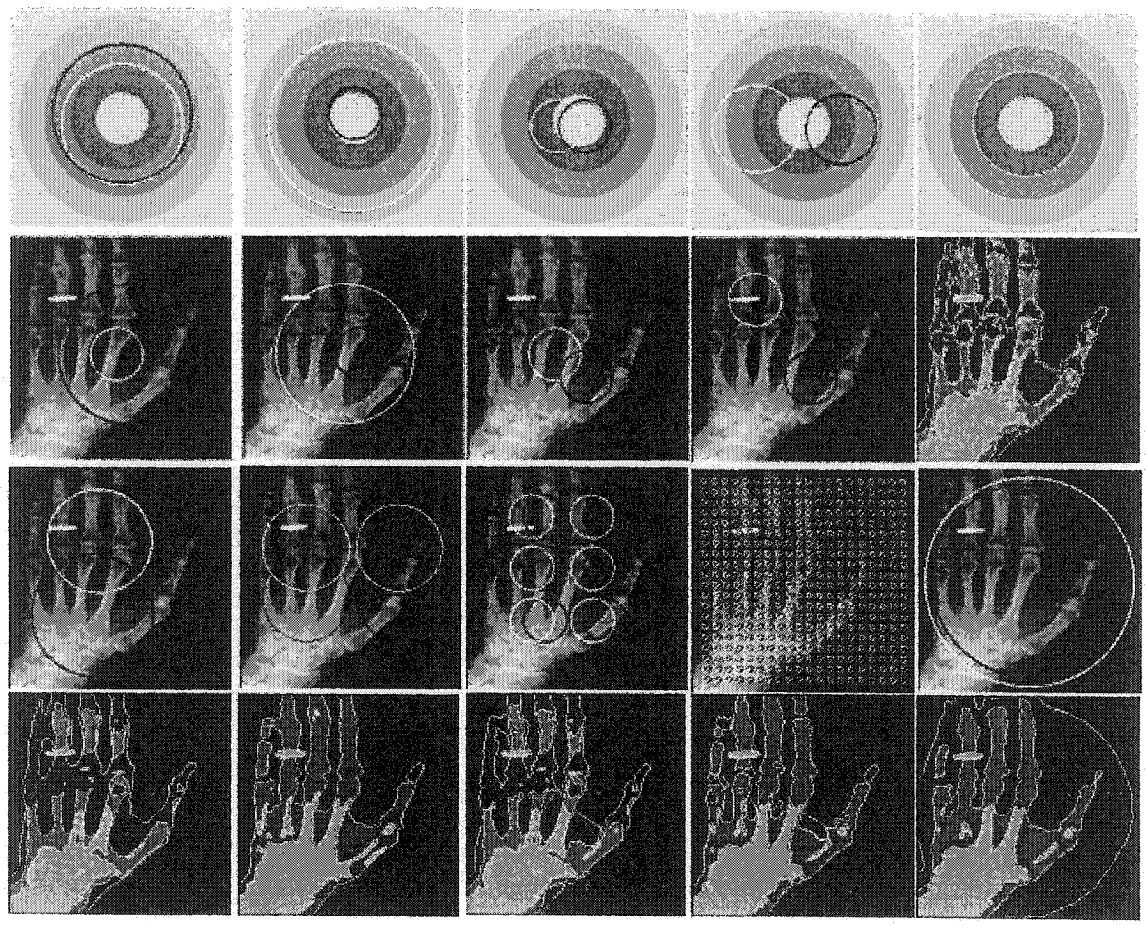


Fig. 6.1. Segmentation results with different initial conditions. 1st and 2nd rows are the results of our hierarchical method: different initial conditions (white curves for ϕ_1 while black curves for ϕ_2) give the same segmentation result (images in the far right column are the segmented results). The third row shows different initial conditions used in the Chan–Vese algorithm, and the last row contains the corresponding segmentation results. The average machine time: synthetic image, hierarchical method: 0:04min, Chan-Vese: 1:10min. *x*-ray image, hierarchical method: 0:24min, Chan-Vese: 1:48min, image size: 256×256.

In the next example, we demonstrate how the proposed hierarchical segmentation method works for real medical images, and we also show the results at the middle stage of the algorithm. The segmentation results of different medical images with multiple distinct regions are shown in Fig. 6.2. For comparison, we present the results obtained for the first stage segmentation (using one level set function ϕ_1) and the final stage (using two level set functions ϕ_1 and ϕ_2). The left column in Fig. 6.2 contains the original images, which are MRI chest, knee, and brain from top to bottom, and the initial curves of ϕ_1 and ϕ_2 superimposed on them. The middle column is the results obtained after the *first* segmenting stage with the piecewise smooth approximation. Although the segmented results show all parts of the images using the piecewise smooth representation, they are actually only two regions divided by the active contour associated with the level set function ϕ_1. In order to get details within each region, we should go to the next stage where the second level set function ϕ_2 is involved into the algorithm. The right column of Fig. 6.2 contains the results of the piecewise smooth segmentation in this stage which is the final stage of the 4–phase case. It can be seen that the final stage with two level set functions ϕ_1 and ϕ_2 given more detailed segmentations than the first stage with only one level set function ϕ_1. In our experimental experience, we can obtain sufficient segmentation results with two level set functions for most real images. For some images with very complicate features or some special type of images, the third level set function may be needed.

This experimental result shows that with this simple choice of initial curves in the 4–phase hierarchical segmentation approach, we can obtain good segmentation results for real medical images.

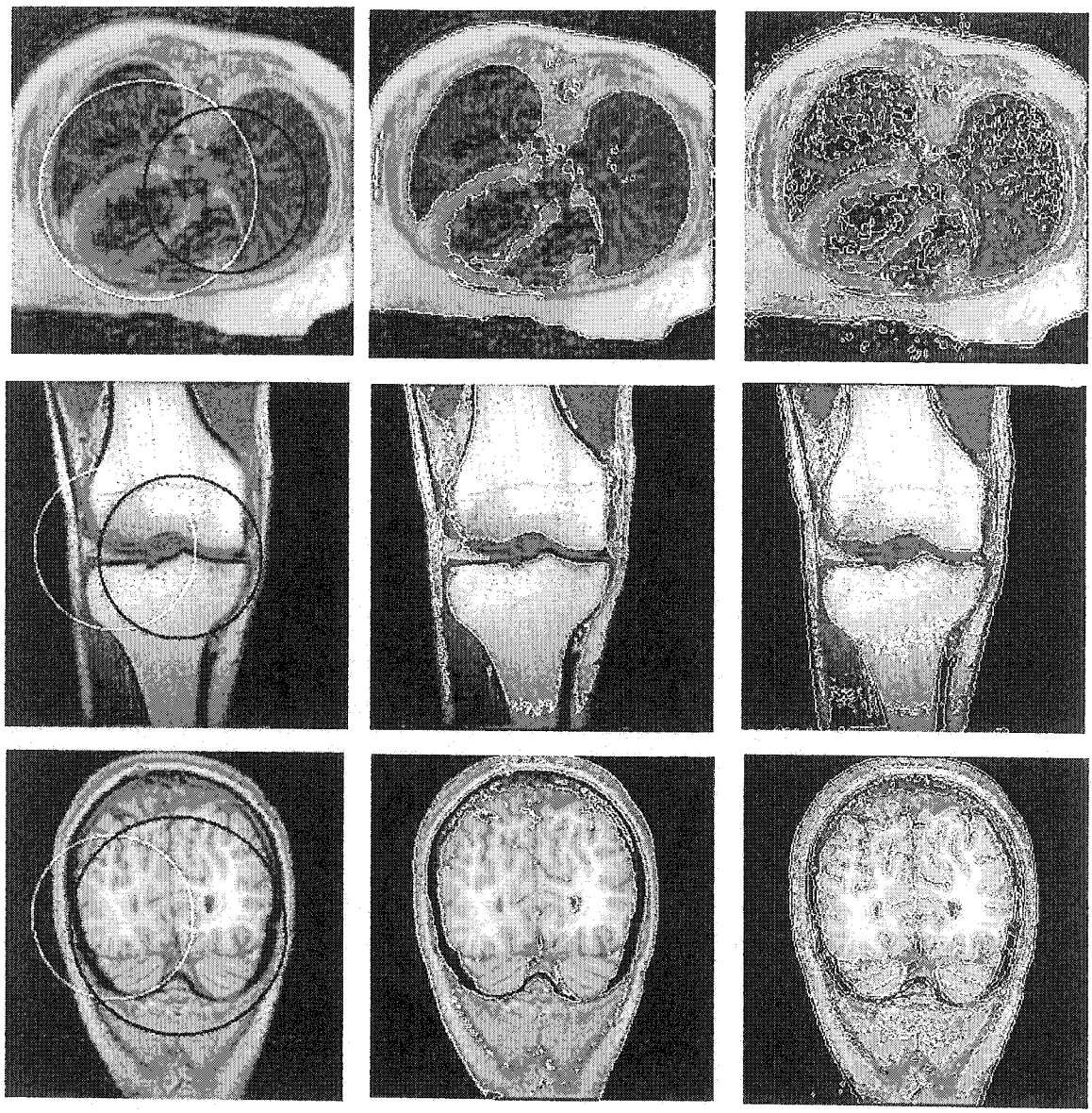


Fig. 6.2. Segmentation of MRI images. Left column: Original images with initial curves, white curves for ϕ_1 while black curves for ϕ_2. Middle column: Results after the first step segmentation while only level set function ϕ_1 involved. Right column: Final segmentations by using two level set functions ϕ_1 and ϕ_2.

6.2 Segmentation and Smoothing Results

We now demonstrate how the proposed two–step segmentation and smoothing algorithm works for the noisy images. Both synthetic images and real medical images with additive Gaussian noise are used in our experiments.

The segmentation and denoising result of a noisy synthetic image with multiple distinct means is shown in Fig. 6.3. In this case, in order to obtain better performance, two level set functions are needed.

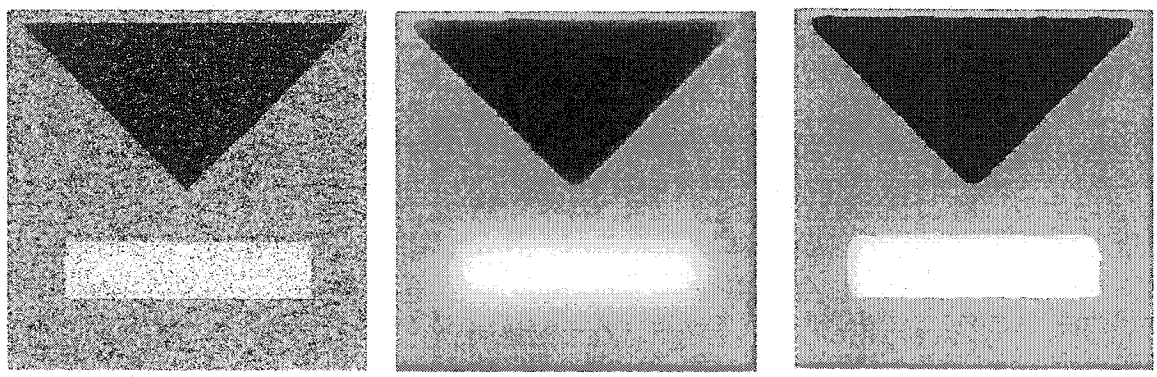


Fig. 6.3. Segmentation and smoothing of a very noisy synthetic image. (a) Image with Gaussian noise (SNR = 10.49 dB). (b) Reconstructed image using the proposed method with 1 level set function. (c) Reconstructed image using the proposed method with 2 level set functions.

For comparisons, we present the denoising result by using one level set function (after the first segmentation stage) in Fig. 6.3(b). We get two regions of the image at that stage: one region is the black triangle, while the white rectangle and the background are assigned to another region. Then we apply our smooth procedure that behaves as an isotropic diffusion in each homogenous region. Since the triangle and the background belong to different regions, the edges of the triangle are preserved as we expected. But

the edges of the rectangle are blurred because it belongs to the same region of the background and diffusion in this region is isotropic. Fig. 6.3(c) shows the result by using two level set functions. Since two different objects and the background are in three different regions and diffusion filtering is applied to each region separately, therefore all edges of the objects are preserved. This experimental result shows that although the algorithm performs linear isotropic diffusion within each homogenous region, the denoising effect for the whole image is similar to anisotropic diffusion filtering.

Medical image segmentation and denoising is an important problem in image processing. In the next experimental example, we demonstrate how the hierarchical multiphase segmentation method is used in the implementation of our selective smoothing algorithm for the segmentation and reconstruction of real medical images. In Fig. 6.4, we present three MRI images with additive Gaussian noise, which are chest, brain and knee from top to bottom in the left column, and the denoised results by using our two-step smoothing method. The segmentation and reconstruction of real MRI images are shown in the middle column. The last column contains the final resulting images with the segmenting curves superimposed on them. It should be noted that the proposed algorithm removes noise very well and preserves and enhances the edges of different regions.

In the last example, we demonstrate how the proposed hierarchical segmentation and selective smoothing method works for a real MRI knee stir image. The original image is shown in Fig. 6.5(a) with initial curves for ϕ_1 and ϕ_2. Fig. 6.5(e) shows the final reconstruction of the image and the segmenting curves superimposed on it. It can be seen from this resulting image some details within the image, such as the white stripe in the

upper part and two small round white matters in the lower part which can not be detected by the Chan–Vese method or our method with $\alpha_1/\alpha_2 = 1$.

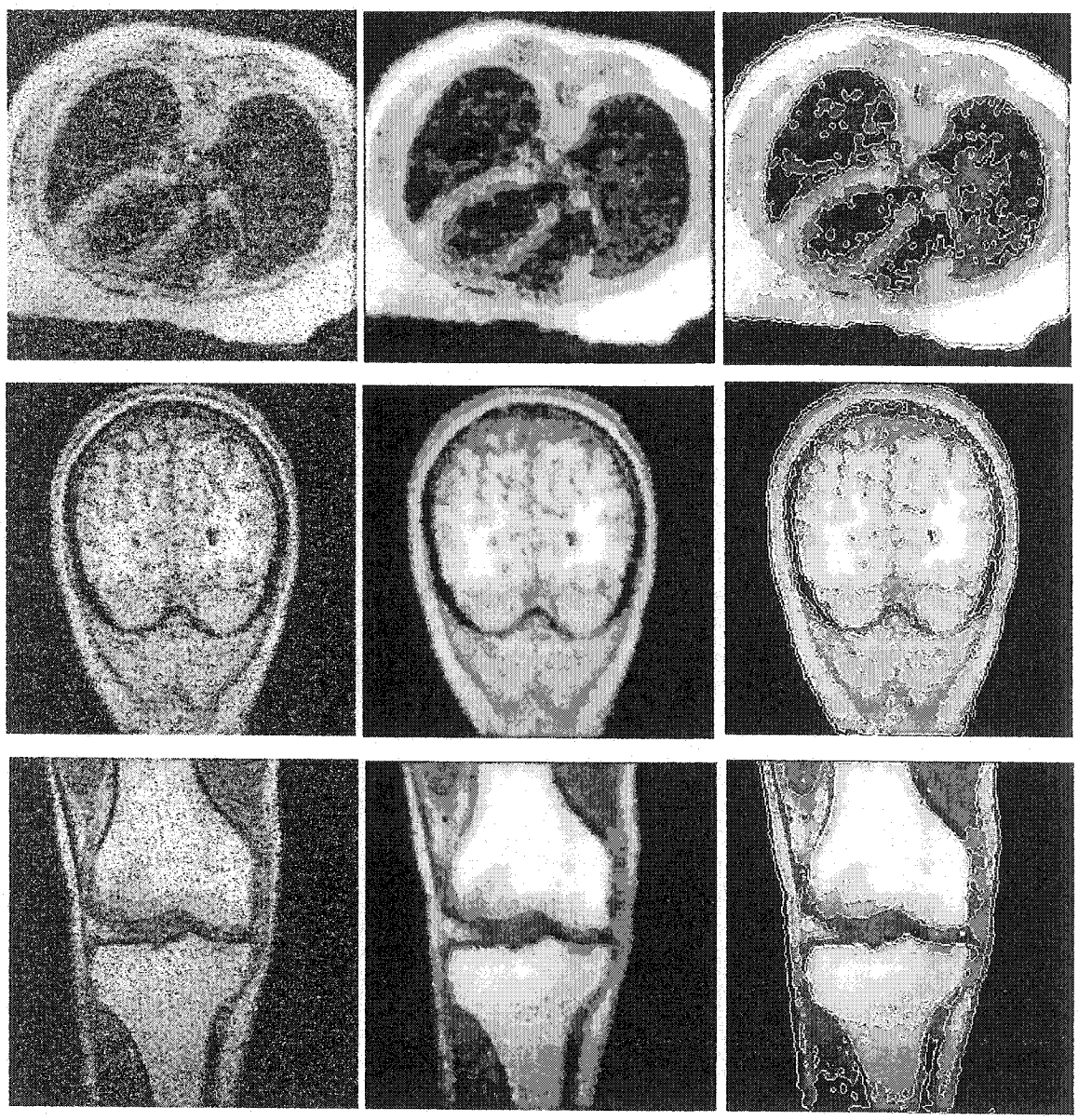


Fig. 6.4. Segmentation and smoothing of noisy MRI images. Left column: Original image with Gaussian noise (*SNR* = 11.06, 10.72, 10.68 dB, from top to bottom). Middle column: Final segmented and reconstructed image based on the proposed method with 2 level set functions. Right column: Segmenting curves are superimposed on final reconstructed images

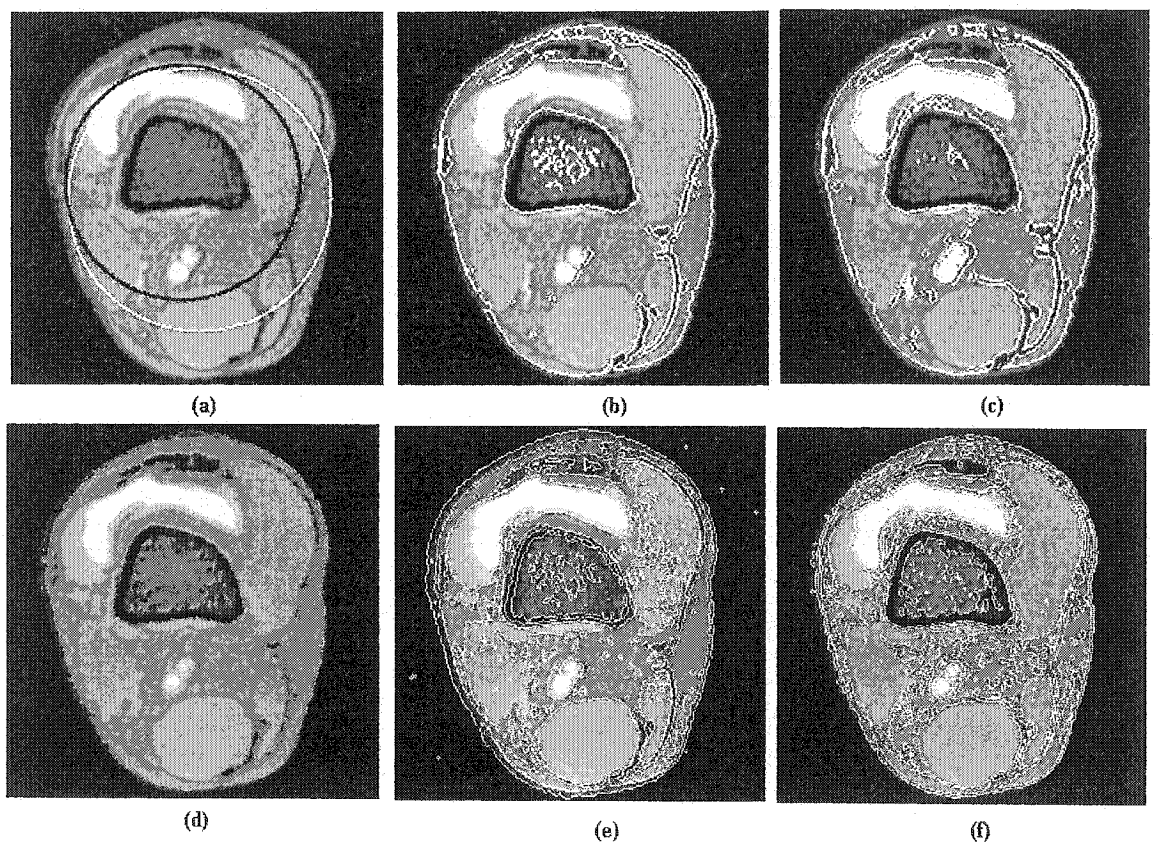


Fig. 6.5. Segmentation and smoothing of real MRI knee stir image. (a) Original image with initial curves, white curve for ϕ_1 and black curve for ϕ_2. (b) and (c) First stage smoothed images with different ratio of α_1/α_2; (b) $\alpha_1/\alpha_2 = 1$, (c) $\alpha_1/\alpha_2 = 3$. (d) Final reconstructed image ($\alpha_1/\alpha_2 = 3$). (e) and (f) Final segmented and reconstructed image with segmenting curves; (e) $\alpha_1/\alpha_2 = 1$, (f) $\alpha_1/\alpha_2 = 3$.

In order to get more details within the region of interest, as discussed in Fig. 4.7 we can choose different values of the weight parameters α_1 and α_2 in the *first* segmentation stage. It is easy to specify the parameter with a simple single initial curve at the *first* stage. For example, for the initial (white) curve as shown in fig. 6.5(a), we can get more details inside the curve by choosing $\alpha_1 > \alpha_2$. Fig. 6.5(d) shows the final segmentation and reconstruction with the choice of $\alpha_1/\alpha_2 = 3$. Fig. 6.5(f) shows the same result of Fig.

6.5(d) with the segmenting curves superimposed. We also present the first stage results in (b) and (c) for comparison.

These figures demonstrate that our segmentation and smoothing method provides better details in different regions, relatively enhances the edges between regions, and highlights the regions of interest.

Chapter 7

Conclusions and Future Work

In this thesis we have proposed and implemented a new image segmentation and smoothing algorithm based on the Chan–Vese active contour model and PDE–based diffusion techniques. The level set method is employed in our numerical implementations. This algorithm works in two steps, first segments the regularized image by using hierarchical piecewise constant segmentation method, then using PDE-based diffusion method smoothes and denoises each segmented region of the original image separately but not across the boundaries.

Because of the coupling of different curve evolution PDEs associated with different level set functions in the Chan–Vese multiphase segmentation algorithm, the initialization of the level set functions becomes a difficult problem. The proposed hierarchical approach decouples the curve evolution PDEs, makes the initialization problem easy to handle, and also speeds up the algorithm.

Compared with the previous simultaneous segmentation and smoothing methods [4], [5], the proposed method is more efficient and flexible. Firstly, we separate the segmentation and smoothing processes, and use hierarchical piecewise constant segmentation algorithm in the segmentation process. Therefore it improves the computational speed drastically and makes the initial condition easy to handle. Secondly, since we use the regularized image obtained by convolving the original noisy image with

a Gaussian filter, edge detection is not sensitive to noise. Thus we can obtain better segmentation of noisy images. Finally, the proposed method allows us to apply different smoothing algorithms in different regions in an image, so it is very convenient when the applications need to highlight some special regions in an image. For example, the inverse diffusion technique [94] can be implemented in our method for edge enhancement. Like anisotropic diffusion methods, the proposed algorithm only smoothes the image within the homogeneous regions but not across the boundaries, thus edges are preserved during the denoising process. The proposed method can process very noisy image with good performance. Our experimental results show that for very noisy images we can still detect the objects and preserve the boundaries of the objects within the image when we remove noise. Actually, this model can perform active contours, denoising, segmentation and edge detection in a unified way.

The segmentation and smoothing method proposed in this thesis can be extended in several directions. One direction is to develop the application specific smoothing algorithms for the purpose of denoising. Since our image segmentation and smoothing method provides a very convenient way for choice of the smoothing algorithm, it is interesting to investigate different kinds of the denoising algorithms that can be used for different purposes of application, for example, the image denoising algorithms using wavelet transformation [88], [95], [96]. The wavelet–based denoising algorithms are very different in technique from the PDE denoising methods, but quite close in spirit and experimental results. Very recently Mrazek et al. studied the connections between the nonlinear diffusion and shift–invariant wavelet shrinkage in one–dimensional case [97]. We may use wavelet based denoising algorithm in our two–step segmentation and

smoothing method for the purpose of denoising. Since the proposed algorithm is based on the level set methods, it can be easily extended to solve the image processing problems in higher dimensional space [92]–[94]. Therefore, another direction is to extend our segmentation and smoothing method to three–dimensional image segmentation and smoothing applications, especially for medical image processing.

Bibliography

[1] R. Wilson, and M. Spann, *Image segmentation and Uncertainty,* John Wiley & Sons Inc. 1988.

[2] R. C. Gonzalez, R. E. Woods, *Digital image processing,* 2nd ed., pp. 567–642, Upper Saddle River, N. J., Prentice Hall, 2002.

[3] D. Mumford, and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math*, vol. 42, pp. 577–685, 1989.

[4] L. Vese and T. F. Chan, "A multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model," *International Journal of Computer Vision* 50(3), pp. 271–293, 2002.

[5] A. Tsai, A. Yezzi, and Alan Willsky, "Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1169–1185, 2001.

[6] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial differential equations and the Calculus of Variations*, Applied Mathematical Sciences vol. 147, Springer-Verlag, 2002.

[7] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1988.

[8] V. Caselles, R. Kimmel and G. Sapiro, "Geodesic Active Contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[9] T. F. Chan and L. A. Vese, "Active Contours without edges," *IEEE transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[10] T. Chan and L. Vese, "An active contour model without edges," Scale-Space Theories in Computer Vision, *Lecture Notes in Computer Science*, no. 1682, pp. 141–151, 1999.

[11] T. Chan, B. Sandberg, and L. Vese, "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, 130–141, June 2000.

[12] T. Chan and L. Vese, "An Efficient Variational Multiphase Motion for the Mumford-Shah Segmentation Model," *Proceedings of the 34th Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 490–494, 2000.

[13] T. Chan, L. Vese, "A level set algorithm for minimizing the Mumford-Shah functional in image processing," *IEEE/Computer Society Proceedings of the 1st IEEE Workshop on "Variational and Level Set Methods in Computer Vision"*, pp. 161–168, 2001.

[14] T. Chan, J. Shen, and L. Vese, "Variational PDE Models in Image Processing," *Notices of the American Mathematical Society*, vol. 50, no. 1, pp. 14–26, Jan., 2003.

[15] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol.12, pp. 629–639, 1990.

[16] F. Catte, P. L. Lions, J. M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numer. Anal.*, vol. 29, no. 1, pp. 182–193, 1992.

[17] L. Alvarez, P. L. Lions, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion II," *SIAM J. Numer. Anal.*, vol. 29, no. 3, pp. 845–866, 1992.

[18] R. Malladi and J. A. Sethian, "Image processing via level set curvature flow," *Proc. Natl. Acad. of Sci. USA,* vol. 92, pp. 7046–7050, July 1995.

[19] R. Malladi and J.A. Sethian, "Image Processing: Flows under Min/Max Curvature and Mean Curvature," *Graphical Models and Image Processing,* vol. 58, no. 2, pp. 127–141, March 1996.

[20] B. B. Kimia and K. Siddiqi, "Geometric heat equation and nonlinear diffusion of shapes and images," *Computer Vision and Image understanding*, vol. 64, no. 3, pp. 305–322, 1996.

[21] J. Weickert, "Theoretical foundations of anisotropic diffusion in image processing," *Computing, Suppl.* 11, pp. 221–236, 1996.

[22] J. Weickert, "Coherence-enhancing diffusion filtering," *International Journal of Computer Vision,* vol. 31, pp. 111–127, 1999.

[23] J. Weickert, B.M. ter Haar Romeny, and M.A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Transactions on Image Processing*, vol. 7, pp. 398–410, 1998.

[24] J. Weickert, and C. Schnörr, "PDE-based preprocessing of medical images," *Künstliche Intelligenz*, no. 3, pp. 5–10, 2000.

[25] J. Weickert, *Anisotropic diffusion in Image Processing*, Teubner, Stuttgart 1998.

[26] S. J. Osher, and J.A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, pp.12–49, 1988.

[27] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999.

[28] S. Osher, and R. P. Fedkiw, "Level set methods: An overview and some recent results," *Journal of Computational Physics*, vol. 169, pp.463–502, 2001.

[29] D. Adalsteinsson, and J. A. Sethian, "A Fast Level Set method for Propagating Interfaces," *Journal of Computational Physics*, vol. 118, no. 2, pp.269–277, 1995.

[30] J. A. Sethian, "Adaptive Fast Marching and Level Set methods for Propagating Interfaces," *Acta Math. Univ. Comenianae*, vol. LXVII, 1, pp. 3–15, 1998.

[31] H. K. Zhao, T. F. Chan, B. Merriman, S. Osher, "A Variational Level Set Approach to Multiphase Motion," *Journal of Computational Physics*, vol. 127, pp.179–195, 1996.

[32] S. Osher, and R. P. Fedkiw, *Level set methods and Dynamic Implicit Surfaces*, Springer Verlag, 2002.

[33] R. P. Fedkiw, G. Sapiro, C. W. Shu, "Shock capturing, level sets, and PDE based methods in computer vision and image processing: a review of Osher's contribution," *Journal of Computational Physics*, vol. 185, pp. 309–341, 2003.

[34] S. Osher, and N. Paragios, (ed.) *Geometric Level Set Method in Imaging, Vision, and Graphics,* Springer, 2003.

[35] G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists*, 5th ed. Academic Press, San Diego, 2001.

[36] H. Goldstein, *Classical Mechanics, 2nd ed.* Reading, MA: Addison-Wesley, 1980.

[37] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C++: the art of scientific computing*, 2nd ed., pp. 829–890, Cambridge University Press, 2002.

[38] D. Betounes, *Partial differential equations for computational science: with Maple and vector analysis,* New York, TELOS, 1998.

[39] J. A. Sethian, "Curvature and Evolution of front," *Communication of Mathematical Physics*, vol. 101, pp. 487–499, 1985.

[40] L. G. Roberts, "Machine perception of three-dimensional solids," *In Optical and Electro-optical Information processing*, edited by J. Tippett, D. Berkowitz, L. Clapp, and A. Vanderburgh, MIT Press, Cambridge, pp. 159–197, 1965.

[41] J. M. S. Prewitt, "Object enhancement and extraction," In *Picture processing and psychopictories*, edited by B. S. Lipkin and A. Rosenfeld, New York, Academic Press, pp. 75–149, 1970.

[42] I. Sobel, "An isotropic 3×3 image gradient operator," in *Machine Vision for Three-Dimensional Scenes*, edited by H. Freeman, Academic Press, pp. 376–379, 1990.

[43] J. Canny, "A computational approach for edges detection," *IEEE Transaction on Pattern Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[44] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *The international Journal of Computer Vision*, vol. 1, no. 2, pp. 167–187, May 1987.

[45] J. Shen and S. Castan, "An optimal linear operator for step edge detection," *CVGIP: Graphics Models and Images Processing,* vol. 54, no. 2, pp. 112–133, March 1992.

[46] J. R. Fram and E. S. Deutsch, "On the quantitative evaluation of edge detection schemes and their comparison with human performance," *IEEE Transaction on Computers*, vol. C-24, no. 6, pp. 616 –628, 1975.

[47] D. Marr, and E. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London,* Series B, 207, pp.187–217, 1980.

[48] R. Haralick, "Digital step edges from zero crossing of second directional derivatives," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 6, no. 1, pp. 58–68, Jan., 1984.

[49] S.R. Gunn, "Edge Detection Error in the Discrete Laplacian of Gaussian," *Proc. IEEE Int. Conf. on Image Processing,* (Chicago, U.S.A.), 1998.

[50] S.R. Gunn, "On the Discrete Representation of the Laplacian of Gaussian," *Pattern Recognition,* vol. 32, no. 8, pp. 1463–1472, 1998.

[51] S. U. Lee, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," *Comput. Vision, Graphics, Image Proc.,* vol. 52, no. 2, pp. 171–190, 1990.

[52] F. H. Y. Chan, F. K. Lam, and H. Zhu, "Adaptive Thresholding by Variational Method," IEEE Trans. Image Proc., vol. 7, no. 3, pp. 468–473, March 1998.

[53] K. S. Fu, and J. K. Mui, "A survey of image segmentation," *Pattern Recognition,* vol. 13, no. 1, pp. 3–16, 1981.

[54] S. A. Hojjatoleslami and J Kittler, "Region growing: A new approach," *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1079–1084, Jul 1998.

[55] A. Jain, *Fundamentals of Digital Image Processing,* Englewood Cliffs, NJ, Prentice Hall, 1999.

[56] S. Wan and W. Higgins, "Symmetric Region Growing," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1007–1015, September 2003.

[57] J. M. Gauch, "Image segmentation and analysis via multiscale gradient watershed hierarchies," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 69–79, Jan., 1999.

[58] S. Mukhopadhyay and B. Chanda," Multiscale Morphological Segmentation of Gray-Scale Images," *IEEE Transactions on Image Processing*, vol. 12, no. 5, pp. 533 –549, 2003.

[59] J. Weickert, "Efficient image segmentation using partial differential equations and morphology," *Pattern Recognition*, vol. 34, no. 9, pp. 1813–1824, September 2001.

[60] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 6, no. 6, pp. 721–741, November 1984.

[61] S. Z Li, *Markov Random Field modeling in Computer Vision*, Springer 1995.

[62] E. Y. Kim, S. H. Park, and H. J. Kim, "A Genetic Algorithm-Based Segmentation of Markov Random Field Modeled Images," *IEEE Signal Processing Letters*, vol. 7, no. 11, pp. 301–303, 2000.

[63] A. Sarkar, M. K. Biswas, and K. M. S. Sharma, "A Simple Unsupervised MRF Model Based Image Segmentation Approach," *IEEE Transactions on Image Processing,* vol. 9, no. 5, pp. 801–812, 2000.

[64] G. Poggi and A. R. P. Ragozini, "Image Segmentation by Tree-Structured Markov Random Fields," *IEEE Signal Processing Letters*, vol. 6, no. 7, pp. 155 –157, 1999.

[65] B. Sumengen, B. S. Manjunath, C. Kenney, "Image Segmentation using Curve Evolution and Flow Fields", *International Conference on Image Processing (ICIP),* Rochester, NY, USA, September 2002.

[66] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "On the evolution of curves via a function of curvature, I: The classical case," *Journal of Mathematical Analysis and Applications,* vol. 163, no. 2, pp. 438–458, January 1992.

[67] D. Terzopoulos, "Deformable Models: Classic, topology-adaptive and generalized formulations," in *Geometric Level Set Method in Imaging, Vision, and Graphics,* pp. 21–40, edited by S. Osher, and N. Paragios, Springer, 2003.

[68] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," In *International Conference on Computer Vision*, pp. 810–815, June, 1995.

[69] G. Aubert, and L. Blanc Féraud, "Some remarks on the equivalence between classical snakes and geodesic active contours," *International Journal of Computer Vision,* vol. 34, no. 1, pp. 19–28, 1999.

[70] A. Bonnet, "On the regularity of the edges set of Mumford–Shah minimizers," *Progress in Nonlinear Differential Equations* vol. 25, pp. 93–103, 1996.

[71] M. Negri and M. Paolini, "Numerical minimization of the Mumford–Shah functional," *Calculus of Variations and Geometric Measure Theory at Pisa,* vol. 38, pp. 67–84, 2001.

[72] J. Shah. "A common framework for curve evolution, segmentation and anisotropic diffusion," In *Proc. IEEE Conference on Computer Version and Pattern Recognition*, xvi+932, pp. 136–142, 1996.

[73] R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 17, no. 2, pp. 158–174, February 1995.

[74] K. Siddiqi, Y. Berub Lauziere, A. Tannenbaum, "Area and Length Minimizing Flows for Shape Segmentation," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 433–443, 1998.

[75] A. Yezzi, A. Tsai, and A. Willsky, "A statistical approach to snakes for bimodal and trimodal imagery," in *Proc. Int. Conf. Computer Vision*, 1999.

[76] G. A. Hewer, C. Kenney and B. S. Manjunath, "Variational image segmentation using boundary functions," *IEEE Transactions on Image Processing*, vol.7, no. 9, pp.1269–1282, 1998.

[77] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum, "A geometric snake model for segmentation of medical imagery," *IEEE Trans. Med. Image*, vol. 16, no. 2, pp. 199 –209, April 1997.

[78] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 884–900, Sept. 1996.

[79] A. Tsai, A. Yezzi, and A. Willsky, "A curve evolution approach to smoothing and segmentation using the Mumford–Shah functional," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 2000.

[80] A. Yezzi, S. Soatto, H. Jin, A. Tsai, and A. Willsky, "Mumford–Shah for segmentation and stereo," in *Geometric Level Set Method in Imaging, Vision, and Graphics*, pp. 207–227, edited by S. Osher, and N. Paragios, Springer, 2003.

[81] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr, "Diffusion Snakes: Introducing Statistical Shape Knowledge into the Mumford-Shah Functional," *International Journal of Computer Vision* 50(3), pp. 295–313, 2002.

[82] Y. Chen et al., "Using Prior Shapes in Geometric Active Contours in a Variational Framework," *International Journal of Computer Vision* 50(3), pp. 315–328, 2002.

[83] A. Tsai, et al., "A Shape-Based Approach to Curve Evolution for Segmentation of Medical Imagery Using Level Sets," *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, pp. 137–154, Feb. 2003.

[84] N. Paragios, "A Level Set Approach for Shape-Driven Segmentation and Tracking of the Left Ventricle," *IEEE Transactions on Medical Imaging*, vol. 22, no. 6, pp. 773–776, June, 2003.

[85] B. Song and T. F. Chan, "A Fast Algorithm for Level Set Based Optimization", UCLA CAM Report 02-68, December 2002.
http://www.math.ucla.edu/~imagers/htmls/reports.html

[86] J. Weickert, "A review of nonlinear diffusion filtering," B. ter Haar Romeny, L. Florack, J. Koenderink, M. Viergever (Eds.), *Scale–Space Theory in Computer Vision, Lecture Notes in Computer Science*, vol. 1252, pp. 3–28, Springer, 1997.

[87] L. Vese, "Multiphase object detection and image segmentation," in *Geometric Level Set Method in Imaging, Vision, and Graphics,* pp. 175–194, edited by S. Osher, and N. Paragios, Springer, 2003.

[88] M.E. Alexander *et al.*, "A wavelet-based method for improving signal-to-noise ratio and contrast in MR images," *Magnetic Resonance Imaging,* vol. 18, pp.169–180, 2000.

[89] N. Paragios, "A Variational Approach for the Segmentation of the Left Ventricle in Cardiac Image Analysis," *International Journal of Computer Vision* 50(3), pp. 345–362, 2002.

[90] T. Chan and L. Vese, "Active Contour and Segmentation Models Using Geometric PDE's for Medical Imaging," in R. Malladi, (Ed.), *Geometric Methods in Bio-Medical Image Processing, Series: Mathematics and Visualization*, Springer, 2002.

[91] G. I. Sanchez-Ortiz, D. Rueckert and P. Burger, "Knowledge-based tensor anisotropic diffusion of cardiac magnetic resonance images," *Medical Image Analysis*, vol. 3, no. 2, pp. 1–25, Oxford University Press, 1999.

[92] D. Magee, A. Bulpitt, and E. Berry, "Level Set Methods for the 3D Segmentation of CT Images of Abnormal Aortic Aneurysms," *Proc. Medical Image Understanding and Analysis* (MIUA), pp. 141–144, July 2001.

[93] T. Preusser and M. Rumpf, " A level set method for anisotropic geometric diffusion in 3D image processing," *SIAM J. Appl. Math.*, vol. 62, no.5, pp. 1772–1793, 2002.

[94] I. Pollak, A. S. Willsky, and H. Krim, "Image Segmentation and Edge Enhancement with Stabilized Inverse Diffusion Equations," *IEEE Transactions on Image processing*, vol. 9, no. 2, pp. 256 –266, 2000.

[95] T. D. Bui and G. Y. Chen, "Translation Invariant Denoising using Multiwavelets," *IEEE Transactions on Signal Processing*, vol. 46, no. 12, pp. 3414–3420, 1998.

[96] G. Y. Chen and T. D. Bui, "Multiwavelets Denoising Using Neighboring Coefficients," *IEEE Signal Processing Letters*, vol. 10, no. 7, pp. 211–214, 2003.

[97] P. Mrazek, J. Weickert, and G. Steidl, "Correspondences between Wavelet Shrinkage and Nonlinear Diffusion," L. D. Griffin and M. Lillholm (Eds.): *Scale–Space 2003, Lecture Notes in Computer Science,* vol. 2695, pp. 101–116, Springer, 2003.

[98] J. W. Thomas, *Numerical Partial Differential equations,* Springer 1995.

Appendix

Numerical Techniques

This chapter concerns the problem of solving numerically the partial differential equations that we have encountered in this thesis. The main numerical techniques have been used in our implementations are the *finite difference methods* [37], [98] for the 2–dimensional parabolic type partial differential equations. The initial and boundary conditions are also considered. Although other kinds of numerical techniques (e.g. finite elements) can be considered, the finite difference methods gain great success in image analysis because the structure of the digital images provide a natural regular grid for discretization. This chapter is an introduction to the two commonly used finite difference methods, namely *Forward Time Centered Space* (*FTCS*) and the *Alternating Directional Implicit* (*ADI*) which are used in the numerical implementations of this thesis.

The prototypical parabolic equation is the diffusion equation, the general form of the diffusion equation in two–dimensional spatial space is,

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\kappa(x, y)\nabla u\right) + f(x, y, t),$$

where κ is the diffusion coefficient, and $f(x, y, t)$ is the source term. The level set equations belong to the category of parabolic equation, so we can use the numerical techniques developed for the diffusion equation for solving the level set equations associated with the curves evolution.

A.1 Finite Difference Numerical Algorithms for the Diffusion Equation

Consider the diffusion equation in 2–dimensional space,

$$u_t = k\left(u_{xx} + u_{yy}\right), \qquad 0 \le x \le L_x,\ 0 \le y \le L_y, \tag{A.1}$$

$$u_x\big|_{x=0} = 0, \quad u_x\big|_{x=L_x} = 0, \quad u_y\big|_{y=0} = 0, \quad u_y\big|_{y=L_y} = 0 \qquad \text{(Boundary conditions)}$$

$$u(x, y, 0) = u_0(x, y), \qquad \text{(Initial condition)}$$

When we solve this equation using the *finite difference* approximation in image processing, it is natural to associate an image with a uniform grid. The grid spacing in the x and y directions is usually chosen equal to $\Delta x = \Delta y = h$. Let Δt be the timestep, $(x_i, y_j) = (ih, jh)$, and $u_{i,j}^n = u(x_i, y_j, n\Delta t)$, while $i = 0, 1, \ldots, M_x - 1$, $j = 0, 1, \ldots, M_y - 1$. Equation (A.1) can be differenced in the obvious way,

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \frac{k}{h^2}\left[\left(u_{i+1j}^n - 2u_{ij}^n + u_{i-1j}^n\right) + \left(u_{ij+1}^n - 2u_{ij}^n + u_{ij-1}^n\right)\right]. \tag{A.2}$$

This is *explicit FTCS* scheme which is first order in time and second order in space, and it is *conditionally stable*. The stability criterion is

$$\Delta t \le \frac{h^2}{4k}\ . \tag{A.3}$$

The *Crank–Nicholson* scheme that is *second–order* accurate in time allows large timestep. If we consider the spatial derivatives on the right–hand side of (A.2) as evaluated at timestep $n+1$, we get the *implicit* (or *backward in time*) scheme as,

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \frac{k}{h^2}\left[\left(u_{i+1j}^{n+1} - 2u_{ij}^{n+1} + u_{i-1j}^{n+1}\right) + \left(u_{ij+1}^{n+1} - 2u_{ij}^{n+1} + u_{ij-1}^{n+1}\right)\right]. \tag{A.4}$$

The *Crank–Nicholson scheme* can be obtained from the average of the explicit and implicit FTCS scheme as,

$$\frac{u_{ij}^{n+1}-u_{ij}^{n}}{\Delta t}=\frac{k}{2(\Delta x)^2}\left[\left(u_{i+1j}^{n+1}-2u_{ij}^{n+1}+u_{i-1j}^{n+1}\right)+\left(u_{i+1j}^{n}-2u_{ij}^{n}+u_{i-1j}^{n}\right)\right]$$

$$+\frac{k}{2(\Delta y)^2}\left[\left(u_{ij+1}^{n+1}-2u_{ij}^{n+1}+u_{ij-1}^{n+1}\right)+\left(u_{ij+1}^{n}-2u_{ij}^{n}+u_{ij-1}^{n}\right)\right]. \qquad \text{(A.5)}$$

This scheme is *unconditionally stable* for linear problems. However, the coefficient matrix needs to be strictly row diagonally dominant and block tridiagonal if we use the natural row ordering. The other problem in this scheme arises in solving the coupled linear equations.

The *alternating directional implicit* (*ADI*) scheme uses a slight different way of generalizing the Crank–Nicholson algorithm, it is still second order in time and space and *unconditionally stable* for linear problems. The idea is to use implicit discretization in one direction while using explicit in another direction and divide each timestep Δt into two steps of size (half–timestep) $\Delta t/2$. The advantages of this method is that each half–step requires only the solution of a simple tridiagonal system, therefore the equations are easier to solve than (A.5) in the Crank–Nicholson scheme. The *ADI* scheme is,

$$\frac{u_{ij}^{n+\frac{1}{2}}-u_{ij}^{n}}{\Delta t/2}=\frac{k}{\Delta x^2}\left(u_{i+1j}^{n+\frac{1}{2}}-2u_{ij}^{n+\frac{1}{2}}+u_{i-1j}^{n+\frac{1}{2}}\right)+\frac{k}{\Delta y^2}\left(u_{ij+1}^{n}-2u_{ij}^{n}+u_{ij-1}^{n}\right), \qquad \text{(A.6a)}$$

$$\frac{u_{ij}^{n+1}-u_{ij}^{n+\frac{1}{2}}}{\Delta t/2}=\frac{k}{\Delta x^2}\left(u_{i+1j}^{n+\frac{1}{2}}-2u_{ij}^{n+\frac{1}{2}}+u_{i-1j}^{n+\frac{1}{2}}\right)+\frac{k}{\Delta y^2}\left(u_{ij+1}^{n+1}-2u_{ij}^{n+1}+u_{ij-1}^{n+1}\right), \qquad \text{(A.6b)}$$

Using the notations $r_x \equiv \frac{k\Delta t}{\Delta x^2}, r_y \equiv \frac{k\Delta t}{\Delta y^2}$; $\delta_x^2 \equiv u_{i-1j}^{n}-2u_{ij}^{n}+u_{i-1j}^{n}$, $\delta_y^2 \equiv u_{ij+1}^{n}-2u_{ij}^{n}+u_{ij-1}^{n}$.

Equations (A.6a) and (A.6b) can be written as,

$$\left(1-\frac{r_x}{2}\delta_x^2\right)u_{ij}^{n+\frac{1}{2}}=\left(1+\frac{r_y}{2}\delta_y^2\right)u_{ij}^{n}, \qquad \text{(A.6a)}'$$

$$\left(1-\frac{r_y}{2}\delta_y^2\right)u_{ij}^{n+1}=\left(1+\frac{r_x}{2}\delta_x^2\right)u_{ij}^{n+\frac{1}{2}}, \qquad \text{(A.6b)}'$$

This method works in two steps: at the first step we use (A.6a)′ to get the result at the first half–timestep $u_{ij}^{n+\frac{1}{2}}$ (from the initial conditions we know u_{ij}^{n}), the second step is to get the solution of the second half–timestep u_{ij}^{n+1} by using (A.6b)′ and $u_{ij}^{n+\frac{1}{2}}$. The process is iterated for each time step.

Step 1) solve equation (A.6a)′

Rewrite (A.6a)′ as,

$$-\frac{r_x}{2}u_{i+1j}^{n+\frac{1}{2}}+(1+r_x)u_{ij}^{n+\frac{1}{2}}-\frac{r_x}{2}u_{i-1j}^{n+\frac{1}{2}}=\frac{r_y}{2}u_{ij+1}^{n}+(1-r_y)u_{ij}^{n}+\frac{r_y}{2}u_{ij-1}^{n} \qquad \text{(A.6a)}''$$

Since we know u^n from the initial condition, the right hand side of (A.6a)″ is constant, denote by r_{ij}. To write the scheme (A.6a)″ in matrix form, we must first consider how to write our vector of solution values. We choose one of the most obvious orderings and set

$$\vec{u}^{\,n}=\left(u_{00}^{n},\cdots,u_{M_x-10}^{n},u_{01}^{n},\cdots,u_{M_x-11}^{n},\cdots,u_{M_x-1M_y-1}^{n}\right)^T.$$

We must realize that the indices started here at 0 and ended at M_x–1 and M_y–1 because we use the Neumann boundary condition at second order (we will discuss the boundary condition in section A.3). This is different from the Dirichlet boundary conditions. For Dirichlet boundary conditions, the indices started at 1 and ended at M_x–2 and M_y–2, because the values with indices of 0, M_x–1 and M_y–1 are not needed.

For each j we have M_x algebraic equations for $i = 0, \ldots, M_x-1$ from (A.6a)″ using a Neumann boundary condition at $x = 0$ and $x = L_x$. The algebraic equations can be written in the matrix–vector form as follows,

$$\begin{pmatrix} 1+r_x & -r_x & 0 & 0 & & & \\ -\frac{r_x}{2} & 1+r_x & -\frac{r_x}{2} & 0 & & & \\ 0 & -\frac{r_x}{2} & 1+r_x & -\frac{r_x}{2} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -\frac{r_x}{2} & 1+r_x & -\frac{r_x}{2} & \\ & & & & -r_x & 1+r_x \end{pmatrix} \begin{pmatrix} u_{0j}^{n+1/2} \\ u_{1j}^{n+1/2} \\ u_{2j}^{n+1/2} \\ \vdots \\ u_{M_x-2\,j}^{n+1/2} \\ u_{M_x-1\,j}^{n+1/2} \end{pmatrix} = \begin{pmatrix} r_{0j} \\ r_{1j} \\ r_{2j} \\ \vdots \\ r_{M_x-2\,j} \\ r_{M_x-1\,j} \end{pmatrix}, \qquad \text{(A.7)}$$

the elements in blank place of the $M_x \times M_x$ matrix are *zero*. This equation can be written in an equivalent vector form as,

$$B\ \vec{u}_j^{\,n+\frac{1}{2}} = \vec{r}_j \qquad \text{(A.7)}'$$

It is a tridiagonal system of algebraic equations for $\vec{u}_j^{\,n+\frac{1}{2}}$. Matrix B and vector $\vec{r}_j$ are known, thus we can use the *Gaussian elimination* method to obtain $\vec{u}_j^{\,n+\frac{1}{2}}$.

With j running from 0 to M_y-1, (A.6a)″ becomes a diagonal system of equations as follows,

$$\begin{pmatrix} B & \Theta & & \\ \Theta & B & & \\ & & B & \\ & & & B \end{pmatrix} \begin{pmatrix} \vec{u}_{j=0}^{\,n+1/2} \\ \vec{u}_{j=1}^{\,n+1/2} \\ \vdots \\ \vec{u}_{j=M_y-1}^{\,n+1/2} \end{pmatrix} = \begin{pmatrix} \vec{r}_{j=0} \\ \vec{r}_{j=1} \\ \vdots \\ \vec{r}_{j=M_y-1} \end{pmatrix}, \qquad \text{(A.8)}$$

where Θ is a $M_x \times M_x$ matrix with all elements are *zero*. Solving the system of equations (A.8) we get $\vec{u}_0^{\,n+1/2}, \ldots, \vec{u}_{M_y-1}^{\,n+1/2}$ which is equivalent to solving M_y systems of (A.7)′.

Step 2) solve equation (A.6b)′

Similar to the procedures in step 1), we rewrite (A.6b)′ as follows,

$$-\frac{r_y}{2}u_{ij+1}^{n+1}+(1+r_y)u_{ij}^{n+1}-\frac{r_y}{2}u_{ij-1}^{n+1}=\frac{r_x}{2}u_{i+1j}^{n+\frac{1}{2}}+(1-r_x)u_{ij}^{n+\frac{1}{2}}+\frac{r_x}{2}u_{i-1j}^{n+\frac{1}{2}}\ . \qquad \text{(A.6b)}''$$

This time fixed i first, we can get a tridiagonal system of algebra equations for $\vec{u}_j^{n+1}$ from (A.6b)″ similar to (A.7) as follows,

$$\begin{pmatrix} 1+r_y & -r_y & 0 & 0 & & & \\ -\frac{r_y}{2} & 1+r_y & -\frac{r_y}{2} & 0 & & & \\ 0 & -\frac{r_y}{2} & 1+r_y & -\frac{r_y}{2} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -\frac{r_y}{2} & 1+r_y & -\frac{r_y}{2} & \\ & & & & -r_y & 1+r_y \end{pmatrix} \begin{pmatrix} u_{i0}^{n+1} \\ u_{i1}^{n+1} \\ u_{i2}^{n+1} \\ \vdots \\ u_{iM_y-2}^{n+1} \\ u_{iM_y-1}^{n+1} \end{pmatrix} = \begin{pmatrix} r'_{i0} \\ r'_{i1} \\ r'_{i2} \\ \vdots \\ r'_{iM_y-2} \\ r'_{iM_y-1} \end{pmatrix}, \qquad \text{(A.9)}$$

and the equivalent form,

$$Q\ \vec{u}_i^{n+1}=\vec{r}_i', \qquad \text{(A.9)}'$$

where r_{ij}' is the right hand side of (A.6b)″ which takes the solutions from step 1). Q is a $M_y \times M_y$ matrix similar to matrix B. The final solutions of u_{ij}^{n+1} can be obtained by solving the following whole system of equations,

$$\begin{pmatrix} Q & \Theta & & \\ \Theta & Q & & \\ & & Q & \\ & & & Q \end{pmatrix} \begin{pmatrix} \vec{u}_{i=0}^{n+1} \\ \vec{u}_{i=1}^{n+1} \\ \vdots \\ \vec{u}_{i=M_x-1}^{n+1} \end{pmatrix} = \begin{pmatrix} \vec{r}'_{i=0} \\ \vec{r}'_{i=1} \\ \vdots \\ \vec{r}'_{i=M_x-1} \end{pmatrix}. \qquad \text{(A.10)}$$

A.2 Numerical Techniques for the Curve Evolution with the Level Set Method

In this section we discuss the discretization of the level set PDEs governing the curve evolution using the finite difference method. We use the following finite difference notations for the level set function ϕ:

$$\Delta_-^x \phi_{i,j} = \phi_{i,j} - \phi_{i-1,j}; \qquad \Delta_+^x \phi_{i,j} = \phi_{i+1,j} - \phi_{i,j}; \qquad \Delta^x \phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2}.$$

$$\Delta_-^y \phi_{i,j} = \phi_{i,j} - \phi_{i,j-1}; \qquad \Delta_+^y \phi_{i,j} = \phi_{i,j+1} - \phi_{i,j}; \qquad \Delta^y \phi_{i,j} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2}.$$

The curvature term κ that plays an important role in the level set PDEs (for example (2.7), and (4.6) etc.) can be approximated by [31],

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = \frac{1}{\Delta x} \left(\frac{\phi_x}{|\nabla \phi|} \right)_x + \frac{1}{\Delta y} \left(\frac{\phi_y}{|\nabla \phi|} \right)_y, \tag{A.11}$$

where, $\phi_x = \frac{1}{\Delta x} \Delta_+^x \phi_{i,j}$, $\left(\frac{\phi_x}{|\nabla \phi|} \right)_x = \frac{1}{\Delta x} \Delta_-^x \left(\frac{\phi_x}{|\nabla \phi|} \right)$, and there are same formulae for y.

Thus,

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right)_{i,j} = \frac{1}{\Delta x} \left[\left(\frac{\phi_x}{|\nabla \phi|} \right)_{i,j} - \left(\frac{\phi_x}{|\nabla \phi|} \right)_{i-1,j} \right] + \frac{1}{\Delta y} \left[\left(\frac{\phi_y}{|\nabla \phi|} \right)_{i,j} - \left(\frac{\phi_y}{|\nabla \phi|} \right)_{i,j-1} \right] \tag{A.12}$$

With this discrete method applied to the curvature term κ we can approximate the curve evolution level set PDEs by using different finite difference schemes (for example FTCS and ADI). The FTCS method was used in the implementations of Chan and Vese's works [4], [9] we use both FTCS and ADI methods in our numerical implementations of the thesis. In order to get stable solutions in the FTCS method we need to use a small

timestep. For the ADI method we can use larger timestep because the solution is unconditionally stable.

A.2.1 The Forward Time Centered Space (FTCS) Method

Let us use the level set PDE (4.6) as an example to describe this numerical approximation method. Following the finite difference schemes discussed above, the discretization of (4.6) can be obtained as,

$$\frac{\phi_{ij}^{n+1}-\phi_{ij}^{n}}{\Delta t}=\delta\left(\phi_{ij}^{n}\right)\left[\frac{\nu}{h^{2}}\left(\frac{\Delta_{+}^{x}\phi_{ij}^{n}}{\sqrt{\frac{1}{h^{2}}\left(\Delta_{+}^{x}\phi_{ij}^{n}\right)^{2}+\frac{1}{(2h)^{2}}\left(\Delta^{y}\phi_{ij}^{n}\right)^{2}}}-\frac{\Delta_{+}^{x}\phi_{i-1j}^{n}}{\sqrt{\frac{1}{h^{2}}\left(\Delta_{+}^{x}\phi_{i-1j}^{n}\right)^{2}+\frac{1}{(2h)^{2}}\left(\Delta^{y}\phi_{i-1j}^{n}\right)^{2}}}\right.\right.$$

$$\left.\left.+\frac{\Delta_{+}^{y}\phi_{ij}^{n}}{\sqrt{\frac{1}{(2h)^{2}}\left(\Delta^{x}\phi_{ij}^{n}\right)^{2}+\frac{1}{h^{2}}\left(\Delta_{+}^{y}\phi_{ij}^{n}\right)^{2}}}-\frac{\Delta_{+}^{y}\phi_{ij-1}^{n}}{\sqrt{\frac{1}{(2h)^{2}}\left(\Delta^{x}\phi_{ij-1}^{n}\right)^{2}+\frac{1}{h^{2}}\left(\Delta_{+}^{y}\phi_{ij-1}^{n}\right)^{2}}}\right)\right.$$

$$\left.-\alpha_{1}\left(u_{0.ij}-c_{1}(\phi_{ij}^{n})\right)+\alpha_{2}\left(u_{0.ij}-c_{2}(\phi_{ij}^{n})\right)\right]. \qquad \text{(A.13)}$$

After doing some algebraic calculations, the solution of the level set equation (4.6) can be written as,

$$\phi_{ij}^{n+1}=\frac{1}{C}\left[\phi_{ij}^{n}+m\left(C_{1}\phi_{i+1j}^{n}+C_{2}\phi_{i-1j}^{n}+C_{3}\phi_{ij+1}^{n}+C_{4}\phi_{ij-1}^{n}\right)\right.$$

$$\left.+\Delta t\delta(\phi_{ij})\left(-\alpha_{1}(u_{0,ij}-c_{1}^{n})+\alpha_{2}(u_{0,ij}-c_{2}^{n})\right)\right], \qquad \text{(A.14)}$$

where, $C= 1+m(C_1+C_2+C_3+C_4)$, $m=\frac{\Delta t}{h^{2}}\delta(\phi_{ij}^{n})\nu$, $\delta(x)=\frac{1}{\pi}\frac{\varepsilon}{\varepsilon^{2}+x^{2}}$, and the coefficients C_1, C_2, C_3, and C_4 can be expressed as follows,

$$C_1 = \frac{1}{\sqrt{\left(\frac{\phi^n_{i+1j} - \phi^n_{ij}}{h}\right)^2 + \left(\frac{\phi^n_{ij+1} - \phi^n_{ij-1}}{2h}\right)^2}}, \qquad C_2 = \frac{1}{\sqrt{\left(\frac{\phi^n_{ij} - \phi^n_{i-1j}}{h}\right)^2 + \left(\frac{\phi^n_{i-1j+1} - \phi^n_{i-1j-1}}{2h}\right)^2}},$$

$$C_3 = \frac{1}{\sqrt{\left(\frac{\phi^n_{i+1j} - \phi^n_{i-1j}}{2h}\right)^2 + \left(\frac{\phi^n_{ij+1} - \phi^n_{ij}}{h}\right)^2}}, \qquad C_4 = \frac{1}{\sqrt{\left(\frac{\phi^n_{i+1j-1} - \phi^n_{i-1j-1}}{2h}\right)^2 + \left(\frac{\phi^n_{ij} - \phi^n_{ij-1}}{h}\right)^2}}.$$

A.2.2 The Alternating Directional Implicit (ADI) Method

As has been discussed in the last section, the ADI method works in two steps, at the *first* half timestep $\Delta t/2$ (from n to $n+\frac{1}{2}$), we do the discretization of (4.6) implicitly in x whereas explicitly in y, thus we have,

$$\frac{\phi^{n+\frac{1}{2}}_{ij} - \phi^n_{ij}}{\Delta t/2} = \delta\left(\phi^n_{ij}\right)\left[\frac{\nu}{h^2}\left(C_1\Delta^x_+\phi^{n+\frac{1}{2}}_{ij} - C_2\Delta^x_+\phi^{n+\frac{1}{2}}_{i-1j} + C_3\Delta^y_+\phi^n_{ij} - C_4\Delta^y_+\phi^n_{ij-1}\right)\right.$$

$$\left. - \alpha_1\left(u_{0,ij} - c_1(\phi^n_{ij})\right) + \alpha_2\left(u_{0,ij} - c_2(\phi^n_{ij})\right)\right], \qquad \text{(A.15)}$$

where C_1, C_2, C_3, and C_4 are the same coefficients as we have defined in section A.2.1.

Let $r_{x1} = \frac{\nu\Delta t}{h^2}\delta\left(\phi^n_{ij}\right)C_1$, $r_{x2} = \frac{\nu\Delta t}{h^2}\delta\left(\phi^n_{ij}\right)C_2$, $r_{y1} = \frac{\nu\Delta t}{h^2}\delta\left(\phi^n_{ij}\right)C_3$, $r_{y2} = \frac{\nu\Delta t}{h^2}\delta\left(\phi^n_{ij}\right)C_4$,

equation (A.15) becomes,

$$-\frac{r_{x1}}{2}\phi^{n+\frac{1}{2}}_{i+1j} + (1 + \frac{r_{x1} + r_{x2}}{2})\phi^{n+\frac{1}{2}}_{ij} - \frac{r_{x2}}{2}\phi^{n+\frac{1}{2}}_{i-1j} = R_{ij}, \qquad \text{(A.16)}$$

where,

$$R_{ij} = \frac{r_{y1}}{2}\phi^n_{ij+1} + (1 - \frac{r_{y1} + r_{y2}}{2})\phi^n_{ij} + \frac{r_{y2}}{2}\phi^n_{ij-1} + \frac{\Delta t}{2}\delta\left(\phi^n_{ij}\right)\left[-\alpha_1\left(u_{0,ij} - c_1(\phi^n_{ij})\right) + \alpha_2\left(u_{0,ij} - c_2(\phi^n_{ij})\right)\right].$$

Using the procedures as discussed in section A.1 and the Neumann boundary conditions

$\frac{\partial \phi}{\partial x} = 0$ for $x = 0$, and $x = L_x$, and $\frac{\partial \phi}{\partial y} = 0$ for y = 0, and $y = L_y$, we can get the solution of $\phi_{ij}^{n+\frac{1}{2}}$ by solving the resulting tridiagonal system of algebraic equations.

For the *second* half timestep (from $n+\frac{1}{2}$ to n+1), we do the discretization of (4.6) explicitly in x and implicitly in y, thus we have,

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^{n+\frac{1}{2}}}{\Delta t/2} = \delta\left(\phi_{ij}^{n+\frac{1}{2}}\right)\left[\frac{\nu}{h^2}\left(C_1\Delta_+^x\phi_{ij}^{n+\frac{1}{2}} - C_2\Delta_+^x\phi_{i-1j}^{n+\frac{1}{2}} + C_3\Delta_+^y\phi_{ij}^{n+1} - C_4\Delta_+^y\phi_{ij-1}^{n+1}\right)\right.$$

$$\left. - \alpha_1\left(u_{0,ij} - c_1(\phi_{ij}^n)\right) + \alpha_2\left(u_{0,ij} - c_2(\phi_{ij}^n)\right)\right]. \qquad \text{(A.17)}$$

Using the same notations, we can write (A.17) as,

$$-\frac{r_{y1}}{2}\phi_{ij+1}^{n+1} + (1 + \frac{r_{y1} + r_{y2}}{2})\phi_{ij}^{n+1} - \frac{r_{y2}}{2}\phi_{ij-1}^{n+1} = \mathrm{R'_{ij}}, \qquad \text{(A.18)}$$

where,

$$R'_{ij} = \frac{r_{x1}}{2}\phi_{i+1j}^{n+\frac{1}{2}} + (1 - \frac{r_{x1} + r_{x2}}{2})\phi_{ij}^{n+\frac{1}{2}} + \frac{r_{x2}}{2}\phi_{i-1j}^{n+\frac{1}{2}}$$

$$+\frac{\Delta t}{2}\delta\left(\phi_{ij}^{n+1/2}\right)\left[-\alpha_1\left(u_{0,ij} - c_1(\phi_{ij}^{n+1/2})\right) + \alpha_2\left(u_{0,ij} - c_2(\phi_{ij}^{n+1/2})\right)\right].$$

Similarly, we can get the final solution of ϕ_{ij}^{n+1}.

A.3 The Neumann boundary conditions

For the two dimensional partial differential equation, assume that the system we want to solve bounded in $0 \le x \le L_x$, $0 \le y \le L_y$, the Neumann boundary conditions in our case can be expressed as,

$$\phi_x(0, y, t) = 0, \quad \phi_x(L_x, y, t) = 0, \quad \phi_y(x, 0, t) = 0, \quad \phi_y(0, L_y, t) = 0 \qquad \text{(A.19)}$$

To approximate the Neumann boundary condition in a manner consistent with our difference scheme, we derive a second order approximation. If we apply the centered difference $\frac{\partial \phi}{\partial x} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}$ at the boundary, the difference operator reaches out of the region. For this reason, we place a *ghost* point, $x_{-1} = -\Delta x$, outside the region, and approximate the boundary condition by

$$\frac{\phi_{1,j}^{n+1} - \phi_{-1,j}^{n+1}}{2\Delta x} = 0. \tag{A.20}$$

This is the second order approximation of the boundary at $x = 0$. For the boundaries at other lines, we can get the condition equations similar to (A.20). We then use these Neumann boundary condition equations to eliminate the terms $\phi_{-1,j}^{n}$, $\phi_{M_x,j}^{n}$, $\phi_{i,-1}^{n}, \phi_{i,M_y}^{n}$ that contain the ghost points.

In the FTCS approach, the boundary condition is very easy to handle. We will give a short discussion of the implementation of the Neumann boundary conditions for the ADI scheme. For the curve evolution level set PDE (4.6), the numerical solution involves solving equations (A.16) and (A.18). Let us examine how the Neumann boundary condition affects the equation (A.16) and (A.18) respectively.

Equation (A.16)

Since the left hand side (*LHS*) of (A.16) only has differences in the x direction, and the right hand side (*RHS*) of (A.16) contains only differences in the y direction, so the Neumann boundary condition at $x = 0$, and $x = L_x$ will affect the left hand side only, whereas the Neumann boundary condition at $y = 0$, and $y = L_y$ affects only the right hand side. Therefore, for the left hand side of (A.16) we have,

if $i = 0$ ($x = 0$), $LHS = \left[1 + \frac{1}{2}(r_{x1} + r_{x2})\right] \phi_{0j}^{n+\frac{1}{2}} - \frac{1}{2}(r_{x1} + r_{x2})\, \phi_{1j}^{n+\frac{1}{2}}$,

if $i = M_x - 1$ ($x = L_x$), $LHS = -\frac{1}{2}(r_{x1} + r_{x2})\phi_{M_x-2\,j}^{n+\frac{1}{2}} + \left[1 + \frac{1}{2}(r_{x1} + r_{x2})\right] \phi_{M_x-1\,j}^{n+\frac{1}{2}}$.

For the right hand side of (A.16) we know,

if $j = 0$ ($y = 0$),

$$RHS = \frac{r_{y1} + r_{y2}}{2}\phi_{i1}^{n} + (1 - \frac{r_{y1} + r_{y2}}{2})\phi_{i0}^{n} + \frac{\Delta t}{2}\delta(\phi_{i0}^{n})\left[-\alpha_1(u_{0,i1} - c_1(\phi_{i0}^{n})) + \alpha_2(u_{0,i0} - c_2(\phi_{i0}^{n}))\right],$$

if $j = M_y - 1$ ($y = L_y$),

$$RHS = \frac{r_{y1} + r_{y2}}{2}\phi_{iM_y-2}^{n} + (1 - \frac{r_{y1} + r_{y2}}{2})\phi_{iM_y-1}^{n}$$

$$+ \frac{\Delta t}{2}\delta(\phi_{iM_y-1}^{n})\left[-\alpha_1(u_{0,iM_y-1} - c_1) + \alpha_2(u_{0,iM_y-1} - c_2)\right].$$

Equation (A.18)

Since the left hand side of (A.18) only has differences in the y direction, and the right hand side of (A.18) contains only differences in the x direction, so the Neumann boundary condition at $y = 0$, and $y = L_y$ will affect the left hand side only, whereas the Neumann boundary condition at $x = 0$, and $x = L_x$ affects only the right hand side. Therefore, for the left hand side of (A.18),

if $j = 0$ ($y = 0$), $LHS = \left[1 + \frac{1}{2}(r_{y1} + r_{y2})\right] \phi_{i0}^{n+1} - \frac{1}{2}(r_{y1} + r_{y2})\phi_{i1}^{n+1}$,

if $j = M_y - 1$ ($y = L_y$), $LHS = -\frac{1}{2}(r_{y1} + r_{y2})\phi_{iM_y-2}^{n+1} + \left[1 + \frac{1}{2}(r_{y1} + r_{y2})\right] \phi_{iM_y-1}^{n+1}$.

For the right hand side of (A.18),

if $i = 0$ ($x = 0$),

$$RHS = (1 - \frac{r_{x1} + r_{x2}}{2})\phi_{0j}^{n+\frac{1}{2}} + \frac{r_{x1} + r_{x2}}{2}\phi_{1j}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\delta\left(\phi_{0j}^{n+\frac{1}{2}}\right)\left[-\alpha_1\left(u_{0,0j} - c_1\right) + \alpha_2\left(u_{0,0j} - c_2\right)\right],$$

if $i = M_x - 1$ ($x = L_x$),

$$RHS = \frac{r_{x1} + r_{x2}}{2}\phi_{M_x-2,j}^{n+\frac{1}{2}} + (1 - \frac{r_{x1} + r_{x2}}{2})\phi_{M_x-1,j}^{n+\frac{1}{2}}$$

$$+ \frac{\Delta t}{2}\delta\left(\phi_{M_x-1,j}^{n+\frac{1}{2}}\right)\left[-\alpha_1\left(u_{0,M_x-1,j} - c_1\right) + \alpha_2\left(u_{0,M_x-1,j} - c_2\right)\right].$$

The implementation of the Neumann boundary condition for the diffusion equation is very similar.

We approximate the initial condition as a circle with center at (x_0, y_0) as,

$$R(i, j) = \sqrt{(ih - x_0)^2 + (jh - y_0)^2}. \tag{A.21}$$

Two initial values are assigned to the level set function ϕ corresponding to the interior region and exterior region of the circle with the radius R_0 as follows,

$$\phi_0(i, j) = \begin{cases} 1 & R(i, j) < R_0, \\ 0 & R(i, j) = R_0, \\ -1 & R(i, j) > R_0. \end{cases} \tag{A.22}$$

In summary, we see that the FTCS scheme is an explicit scheme, and easy to implement. Since it is *conditionally* stable, we cannot make the timestep Δt too large. The *ADI* scheme is a second order scheme that is *unconditionally* stable. The solution procedure in this approach involves solving tridiagonal matrices that are more expensive than using the explicit scheme. If the larger timestep Δt made available for the required accuracy, the ADI scheme is more efficient to use than the explicit scheme.