IMAGE SEGMENTATION AND SMOOTHING VIA PARTIAL DIFFERENTIAL EQUATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

NESLİHAN ÖZMEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

FEBRUARY 2009

Approval of the thesis:

IMAGE SEGMENTATION AND SMOOTHING VIA PARTIAL DIFFERENTIAL EQUATIONS

submitted by **NESLİHAN ÖZMEN** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Scientific Computing, Middle East Technical University** by,

Prof. Dr. Ersan Akyıldız ________________
Dean, Graduate School of **Applied Mathematics**

Prof. Dr. Bülent Karasözen ________________
Head of Department, **Scientific Computing**

Prof. Dr. Kemal Leblebicioğlu ________________
Supervisor, **Electrical and Electronics Engineering, METU**

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu ________________
Electrical and Electronics Engineering, METU

Prof. Dr. Bülent Karasözen ________________
Scientific Computing, METU

Prof. Dr. Gerhard Wilhelm Weber ________________
Scientific Computing, METU

Assist. Prof. Dr. Hakan Öktem ________________
Scientific Computing, METU

Assist. Prof. Dr. İlkay Ulusoy ________________
Electrical and Electronics Engineering, METU

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: NESLİHAN ÖZMEN

Signature :

ABSTRACT

IMAGE SEGMENTATION AND SMOOTHING VIA PARTIAL DIFFERENTIAL EQUATIONS

Özmen, Neslihan
M.S., Department of Scientific Computing
Supervisor : Prof. Dr. Kemal Leblebicioğlu

February 2009, 86 pages

In image processing, partial differential equation (PDE) based approaches have been extensively used in segmentation and smoothing applications. The Perona-Malik nonlinear diffusion model is the first PDE based method used in the image smoothing tasks. Afterwards the classical Mumford-Shah model was developed to solve both image segmentation and smoothing problems and it is based on the minimization of an energy functional. It has numerous application areas such as edge detection, motion analysis, medical imagery, object tracking etc. The model is a way of finding a partition of an image by using a piecewise smooth representation of the image. Unfortunately numerical procedures for minimizing the Mumford-Shah functional have some difficulties because the problem is non convex and it has numerous local minima, so approximate approaches have been proposed. Two such methods are the Ambrosio-Tortorelli approximation and the Chan-Vese active contour method. Ambrosio and Tortorelli have developed a practical numerical implementation of the Mumford-Shah model which based on an elliptic approximation of the original functional. The Chan-Vese model is a piecewise constant generalization of the Mumford-Shah functional and it is based on level set formulation. Another widely used image segmentation technique is the “Active Contours

(Snakes)” model and it is correlated with the Chan-Vese model. In this study, all these approaches have been examined in detail. Mathematical and numerical analysis of these models are studied and some experiments are performed to compare their performance.

Keywords: Image Processing, Partial Differential Equations, Optimization, Functional Analysis, Numerical Analysis.

ÖZ

KISMİ TÜREVLİ DENKLEMLERLE GÖRÜNTÜ BÖLÜTLEME VE DÜZGÜNLEŞTİRME

Özmen, Neslihan
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Prof. Dr. Kemal Leblebicioğlu

Şubat 2009, 86 sayfa

Kısmi türevli denklem (KTD) tabanlı yaklaşımlar görüntü işlemede, bölütleme ve düzgünleştirme uygulamalarında yaygın olarak kullanılmaktadır. Perona-Malik doğrusal olmayan difüzyon modeli görüntü düzgünleştirme alanında kullanılan ilk KTD tabanlı yöntemdir. Bunun ardından klasik Mumford-Shah modeli hem görüntü bölütleme hem de düzgünleştirme problemlerine yönelik olarak geliştirilmiştir ve enerji fonksiyonelinin minimize edilmesini temel alır. Bu modelin kenar bulma, hareket analizi, medikal görüntüleme, nesne izleme gibi bir çok uygulama alanı bulunmaktadir. Bu model, görüntünün parçali ve düzgün temsilini kullanarak bir görüntü bölütlemesi bulmanin bir yoludur. Ne yazık ki, problemin konveks olmaması ve çok sayıda yerel minimum değere sahip olmasından dolayı Mumford-Shah fonksiyonelini minimize edecek nümerik yöntemler bazı zorluklar taşırlar, bu yüzden yaklaşık yöntemler önerilmiştir. Bu yöntemlerden ikisi Ambrosio-Tortorelli yaklaşımı ve Chan-Vese aktif çevrit yöntemidir. Ambrosio ve Tortorelli, Mumford-Shah fonksiyoneli için eliptik yaklaşıma dayalı, pratik bir sayısal uygulama önermişleridir. Chan-Vese modeli, Mumford-Shah fonksiyonelinin parçalı sabit bir genelleştirmesidir ve düzey kümesi formülasyonuna dayanır. Yaygın olan diğer bir görüntü bölütleme tekniği ise "Aktif Çevrit (Yılan)" modelidir ve Chan-Vese modeliyle ilintilidir. Bu çalışmada bütün bu modeller detaylı olarak incelenmiştir. Bu modellerin matematiksel ve sayısal analizleri çalışılmış ve performanslarını karşılaştırmak için denemeler gerçekleştirilmiştir.

Anahtar Kelimeler: Görüntü İşleme, Kısmi Türevli Denklemler, Optimizasyon, Fonksiyonel Analiz, Sayısal Analiz.

To my family

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Prof. Dr. Kemal Leblebicioğlu, for patiently guiding, motivating and encouraging me during this study. I kindly thank Prof. Dr. Gerhard Wilhelm Weber for some very helpful conversations and for suggesting some of the references. I also thank Assist. Prof. Dr. İlkay Ulusoy for her friendship and support throughout this study. I would like to give special thanks to Metin Burak Altınoklu for his help and valuable contribution to this study. I am grateful to my friend Utkan Eryılmaz for his sharing this hard period with me, suggestions, understanding, patience and guidance. I would like thank to Derya Altıntan for her advise and encouragement. I am also thankful to Selime Gürol, who guided me about the direction of my thesis from the beginning. I am thankful to my friends Fatma Yerlikaya and Deniz Yılmaz for being me all the time and motivating me. Moreover, the environment of Computer Vision and Intelligent Systems Research Lab. itself has an important place in this study. I want to thank each of the Lab members, Tülay Akbey, Yasemin Özkan Aydın, Örsan Aytekin and Mehmet Yılmaztürk. My greatest debts are to my family to my father Celal, my mother Suzan, my sister Pelin and my brother Alper for their love, endless support and encouragement. Thanks also to my cousin, Nurcan Gençaslan, for her kind help during this study. This work was supported by The Scientific and Technical Research Council of Turkey MSc. Scholarship Program (TUBITAK Yurt İçi Yüksek Lisans Burs Programı).

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Images in our world have a great meaning and they are extensively used in many tasks in daily basis such as surveillance, identification and communication. Great improvements and technological developments in computer industry allow us to obtain and work with more image data and this progress triggered birth of a new application field known as Computational Imaging [51]. Computational Imaging is interested in acquisitioning and processing the visual information and it is divided into two main application areas: image processing and computer vision. These two fields differ with respect to the type of information used in the process and output obtained after the process. In image processing the processed image is used by the humans, while in computer vision it is used by the computers. Furthermore, computer vision tends to focus on the 3D scene projected onto one or several images, whereas image processing tends to focus on 2D images.

An image is considered as a 2D simulation of the 3D world by image processing community. The images obtained by digital cameras are considered on a discrete grid. Each square in this grid, known as a pixel, has a digital value. Therefore, a digital image is defined as a matrix and each matrix element takes the corresponding pixel value. The number of rows and columns give the size of an image and it is called as resolution. The resolution demonstrates the closeness between the real world and the image [4]. Mathematically an image is expressed by a real-valued, bounded function defined on an open subset of $\Re^2$ and the value of the function at a point gives the intensity or brightness, at that point in the image.

The role of image processing is to develop techniques to extract and exploit information from images. The application areas are numerous and the most significant application fields are medical [10, 37, 60, 62], defense [25, 26, 30] and manufacturing [15, 28, 33, 58]. Image

Processing is comprised of several branches such as image segmentation, image denoising, image registration etc. Each area is used for different purposes. For instance, the ambition of segmentation is to partition an object of interest from the background in a given image. On the other hand, denoising can be defined as recovering the original image from a noisy measurement. The researches on the subject of Image Processing have been extensively carried out for a long time. When we look at the history of image processing, to process the images, several mathematical approaches have been emerged and the most fundamentals among them are stochastic modeling [23, 24, 34], wavelets [8, 16, 17, 35, 63], and partial differential equation (PDE) approaches. Stochastic modeling depends on the theory of Markov random field and wavelet theory is based on decomposition techniques. However, PDE-based approach is undoubtedly the most famous one due to its powerful mathematical background and flexibility [4].

PDEs are generally used for solving the problems of engineering, physics and other scientific applications because they are directly related to real-world problems. They are described in a continuous setting and supply a better comprehension of the discrete world. One can develop stable and novel algorithms of image processing based on PDEs which have a well-established theory for years. Because of this well-established theory behind them and evolutionary characteristics, PDEs are the most practical and preferred tools in image processing. After they were introduced to this area at the beginning of the 80's, several PDE-based methods have emerged. These methods have been successfully used to solve the main problems in various application areas such as image segmentation, edge detection, image deblurring, object tracking, noise removing, image enhancement, image restoration, image inpainting, shape extraction and recognition, motion analysis etc. [2, 5, 13, 46, 47, 48, 54, 56].

Variational techniques are extensively used for solving image processing problems especially by the PDE-based methods. A problem defined by a well chosen energy functional to be minimized is usually equivalent to a nonlinear PDE. The basic idea behind them is to find the local minimum of a given energy functional while the variables correspond to the solution of the problem or the results of a given task. Local optimum of the energy functional can be found by equated its first variation to zero. Numerically iterative optimisation techniques such as method of steepest descent are used to determine the local optimums of energy functional.

One of the earlier classical method used in image processing is "Gaussian Smoothing". It is based on convolution of an image with the Gaussian function

$$G_\sigma(x) = \frac{1}{(4\pi\sigma)^{\frac{d}{2}}} e^{-|x|^2/4\sigma} \ .$$

Researcer have discovered the fact that the Gaussian function is a fundamental solution of a linear heat equation

$$I_t = c\nabla^2 I = cI_{xx} + cI_{yy} \ .$$

This is how PDEs entered into the image processing computations. Later "linear diffusion equation method" was used in many applications. This linear image smoothing method blurs edges in the images and sometimes moves their locations. However; for some applications, it is very crucial to find the location of edges precisely, for instance in the medical imagery, finding a tumor's place exactly is very vital for surgery operations. On the other hand, for some of the applications blurring is not desired, so Gaussian smoothing (linear diffusion) method is a bad choice. To overcome these difficulties, nonlinear diffusion methods are proposed [39].

Perona and Malik [44] have introduced a modification of the linear heat equation, known as anisotropic diffusion, as shown below, which is used in image denoising and smoothing applications.

$$I_t = \nabla \cdot (g(\nabla I)\nabla I) \ ,$$

where $\Omega \subset \Re^2$ is image domain, $I : \Omega \rightarrow \Re$ represents the image function.

This method preserves the edges better than linear diffusion models. Since than this nonlinear diffusion approach has become a center of attraction and many different applications have emerged.

Mathematically, the Perona-Malik (PM) model can be considered as the backward heat equation, it acts locally like heat equation; the regions in the image are diffused with respect to intensities. The function g is considered as an edge indicator and it depends on the gradient ∇I of the image I. g is chosen such that for large gradients it is expected to go to zero. As a result diffusion process is expected to stop on edges. The extensive explanation and mathematical analysis of this model is given in Chapter 2.

Despite its good properties, the PM model is not adequate for some applications. It is an ill-posed problem and have some difficulties such as non-uniqueness of its solutions. Many different approaches have been proposed to overcome the problems of the PM model. The most commonly used one is the regularized version of it originally suggested by Catte, Lions and Morel [7] is as follows

$$I_t = \nabla \cdot (g(|\nabla G_\sigma * I|)\nabla I) \ .$$

In this regularized version, the convolution of the image with the Gaussian kernel is used and this is similar to the linear diffusion, so this model can be considered joining of ideas in linear and nonlinear diffusions. This little contribution have provided favorable benefits and facilitated the verification, the existence and the uniqueness of the solution. Moreover, the convolution of images with the Gaussian kernel ensures quite a smooth function for which its gradient is mathematically easier to calculate.

Alvarez, Lions and Morel [1] developed a nonlinear diffusion equation for image and shape analysis based on mean curvature motion as represented below

$$I_t = g(|\nabla G_\sigma * I|)\nabla I \cdot \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right) \ .$$

In this method the regions are smoothed by the mean curvature motion, while the edges are preserved and comparatively enhanced. The function f, given in [1], is a contrast function and it decides whether a detail is sharp enough to be kept.

Another widely used PDE-based method is proposed by Mumford and Shah in 1989 [41]. They introduced the following energy functional, known as Mumford-Shah (MS) functional

$$E_{MS}(u, C) = \alpha \int\limits_{\Omega - C} |\nabla u|^2 \mathrm{d}x + \beta \int\limits_{\Omega} (u - I)^2 \mathrm{d}x + \mu.\mathrm{length}(C) \ ,$$

where $u : \Omega \rightarrow \Re$ is an approximation of the given image I and C is the set of edges.

This functional is used for solving both image segmentation and denoising problems. The fundamental idea behind this functional is to obtain a piecewise smooth approximation of a given image I and an edge set Γ. In the functional, the first term represents the stabilizing cost, the second term is the deviation cost, and the third term gives the edge cost. To find piecewise smooth approximation u of the image I, the energy functional, consists of these three costs should be minimized. However, this functional has some numerical complications due to the lack of uniqueness of the solution and the dependency of the region integration on its solution. Numerous approximate approaches are proposed to overcome these difficulties. Two such methods are the Ambrosio-Tortorelli (AT) [3] approximation and the Chan-Vese (CV) active contour method [9]. Ambrosio and Tortorelli developed a practical numerical implementation of the Mumford-Shah functional and it is based on an elliptic approximation of the original functional.

$$E_{AT}(u,v) = \int\limits_{\Omega} \left(\alpha(v^2|\nabla u|^2) + \beta(u-I)^2 + \frac{1}{2}\left(\rho|\nabla v|^2 + \frac{(1-v)^2}{\rho}\right)\right) \mathrm{d}x \; .$$

The Chan-Vese model is a piecewise constant generalization of the Mumford-Shah functional and it is based on level set formulation introduced by Osher and Sethian [42].

$$\begin{aligned} E_{CV}(C, c_1, c_2) &= \mu.\mathrm{length}(C) + \nu.\mathrm{area}(inside(C)) \\ &\quad + \lambda_1 \int\limits_{inside(C)} |I - c_1|^2 \mathrm{d}x + \lambda_2 \int\limits_{outside(C)} |I - c_2|^2 \mathrm{d}x \; . \end{aligned}$$

Besides, Tsai et al. [50] suggested an image segmentation and smoothing method based on the Mumford-Shah functional and it resembles to the Chan-Vese model in many respects.

The Snakes (Active Contours) are also image segmentation techniques based on variational methods, originally proposed by Kass et al. [32]. They can be understood as a special case of a more general technique of matching a deformable model to an image by means of energy minimization. In general, snakes are curves defined in an image domain and they move under the control of two image forces: internal force and external force. They are successfully used in medical imaging, object tracking, edge detection and video surveillance applications.

After applying any segmentation algorithm, one may desire to know the accuracy of the results. In general, segmentation algorithms are evaluated visually, qualitatively. However, it is very important to verify the segmentation algorithms by using quantitative measures. So, it is important to develope a metric to determine quality of the segmentation. In [29], a quantitative analysis is carried out to assess the segmentation algorithms. First the ground truth of the image of interest is obtained by manual description, and then a percentage error is chosen to measure the distance between ground truth and segmented image. Binary image representation of the ground truth and segmented image are used to find the error measure and segmentation techniques are compared according to this error. The errors which are produced by segmentation techniques are calculated, and the method which gives the smallest segmentation error is determined as the most accurate technique. In [21], three basic methods, laws of optics application, shape coefficients and template matching, of image segmentation verification are introduced. In the first two methods, an object of known shape and dimension is used as a reference sample and the sizes of reference sample and the segmented object are compared. In the third method, a template, the low quality of image of the plot paper, is used and the consistencies between the image after the segmentation and the template are analyzed. In University of California, Berkeley, a group studying on segmentation and edge detection provides a dataset (Berkeley Segmentation Dataset and Benchmark) which is used for developing and evaluating the segmentation algorithms. The group compares the segmentation results with the ground truth (the human segmented image of the original). They use precision-recall curves to determine similarities between segmentation and ground truth. Precision and recall signify the noise measures in the output of the segmentation and the ground truth, respectively. The details are given in [38].

In this thesis image segmentation methods based on variational techniques including the snakes and the PDE-based methods such as the Perona-Malik, the Mumford-Shah, the Ambrosio-Tortorelli and the Chan-Vese models are examined. The accuracy of these methods is not investigated in quantitative point of view but we qualitatively analyze the segmentation results, namely they are evaluated by means of visual inspection. The organization of this work is as follows: in Chapter 2, analytical properties and numerical approximation of the Perona-Malik model are explained and some implementations of the model are presented. In Chapter 3, the detailed mathematical analysis and problems of the Mumford-Shah functional are given. Analytical and numerical analysis, also experimental results of the the Ambrosio-Tortorelli approximation of the Mumford-Shah functional are discussed in Chapter 4. In Chaper 5, the Chan-Vese active contours model is examined. The mathematical analysis and numerical approximations are presented, and experimental results are demonstrated. In Chapter 6, the technique of active contours are expressed. An overview of this methods is explained and the detailed analysis of the original snakes is given. Chapter 7 includes comparasions of the techniques discussed in previous chapters Finally, in Chapter 7 the conclusions are provided.

CHAPTER 2

PERONA-MALIK MODEL

Partial differential equations are successfully used in many image smoothing tasks. In diffusion processes, the main goal is to remove the noise from the given image and finding the details by smoothing the interior of the regions, not the edges. The original image can be considered as an initial condition of a diffusion process. When evolution by diffusion is applied on the whole image, it is defined as a “scale space” approach and it can be used for extracting important information [14]. Generally, denoising is performed by convolving the original image with a Gaussian function, or equally by solving the heat diffusion equation with the original image as the initial state. During this process the edges get blurred too much and they are not preserved; this is the main disadvantage of the diffusion method. This linear heat diffusion equation is defined as

$$I_t = c\nabla^2 I = cI_{xx} + cI_{yy} \ , \tag{2.1}$$

where c is a unitary constant.

To overcome the difficulties of the linear diffusion and for keeping the edges in an image sharp, well-defined and localized, Perona and Malik proposed a nonlinear diffusion scheme [44]. They introduced a nonlinear heat equation with a diffusion coefficient g. The image domain is defined by Ω and $\partial\Omega$ is the boundary of Ω. The Perona-Malik (PM) model is defined as

$$\frac{\partial I}{\partial t} = div(g \cdot \nabla I) \ , \tag{2.2}$$

$$\frac{\partial I}{\partial n} = 0 \ . \tag{2.3}$$

Notice that, when g is taken as a constant, again the linear heat diffusion equation is obtained. The scalar diffusivity $g(.)$ is not constant; it is defined as a function of the gradient I, and it is a decreasing and non-negative function. In the regions where the gradient is small in magnitude, the diffusion is almost linear. At the regions where the norm of the gradient is large, it restrains the diffusion near the edges; thus the edges are preserved. The selection of the function g is the fundamental characteristic of this method. Perona and Malik proposed two different g functions:

$$g\left(\nabla I\right) = exp\left(-\frac{|\nabla I|}{k^2}\right) ,$$

$$g\left(\nabla I\right) = -\frac{1}{\sqrt{1+\frac{|\nabla I|^2}{k^2}}} ,$$

where k is a fixed gradient threshold.

This diffusion process smoothes the edges with small gradients and the image is divided into two regions along a steep gradient region. The condition to maintain the edges is defined as $|\nabla I| > k$; so the parameter k must decrease when the norm of the gradient decreases. When the diffusion process is allowed for a long time, it blurs the image and some edges can be lost. By using p-norm based estimation of the parameter k, the error descends slower than using a constant k. The parameter k affects the results directly. A method is proposed in [53] for estimating the appropriate k when the suitable value of the diffusion coefficient is not known and loss of details is not wanted. For setting k, the smoothing property of the diffusion equation is used. Since a fixed k may delete edge and small details during the diffusion process, k is decreased for each iteration without noise estimation. The formulation for estimating the parameter k given in [53] is

$$k(m\Delta t) = \frac{\varsigma|\nabla I|_p}{M \times N} , \tag{2.4}$$

where I is an image with M rows and N columns and ς is a constant proportional to the image average intensity. A p-norm of the image is used as defined below

$$|\nabla I|_p = \left[\sum |I|^p\right]^{1/p} . \tag{2.5}$$

2.1 Numerical Approximations

The model is

$$\begin{aligned} \frac{\partial I(x,y,t)}{\partial t} &= div(g(x,y,t)\cdot\nabla I(x,y,t))\ , & (2.6)\\ \frac{\partial I(x,y,t)}{\partial n} &= 0\ . & (2.7)\end{aligned}$$

The initial condition is given by the original image, i.e., $I(x,y,0) = I_0(x,y)$. "Neumann type boundary conditions" can be used as boundary conditions.

The function $I^n_{i,j} = I(ih, jh, n\Delta t)$ is the approximation of the image $I(x,y,t)$ and $g^n_{i,j} = g(ih, jh, n\Delta t)$ is the approximation of the function $g(x,y,t)$. Discretization of the model is

$$\frac{I^{n+1}_{i,j} - I^n_{i,j}}{\Delta t} = div(g\cdot(\nabla I))^n_{i,j}\ , \tag{2.8}$$

$$\nabla I_{i,j} = \begin{pmatrix} \frac{I_{i+\frac{1}{2},j} - I_{i-\frac{1}{2},j}}{h} \\ \\ \frac{I_{i,j+\frac{1}{2}} - I_{i,j-\frac{1}{2}}}{h} \end{pmatrix},$$

for h=1

$$(g\cdot\nabla I)_{i,j} = \begin{pmatrix} g_{i,j}(I_{i+\frac{1}{2},j} - I_{i-\frac{1}{2},j}) \\ \\ g_{i,j}(I_{i,j+\frac{1}{2}} - I_{i,j-\frac{1}{2}}) \end{pmatrix},$$

$$\begin{aligned}(div\,(g\cdot\nabla I))_{i,j} &= g_{i+\frac{1}{2},j}I_{i+1,j} - g_{i-\frac{1}{2},j}I_{i,j} - g_{i+\frac{1}{2},j}I_{i,j} - g_{i-\frac{1}{2},j}I_{i-1,j}\\ &\quad + g_{i,j+\frac{1}{2}}I_{i,j+1} - g_{i,j-\frac{1}{2}}I_{i,j} - g_{i,j+\frac{1}{2}}I_{i,j} + g_{i,j-\frac{1}{2}}I_{i,j-1}\ . \qquad (2.9)\end{aligned}$$

Perona and Malik proposed a numerical approximation for g, as given below:

$$g_{i+\frac{1}{2},j} \approx g_{i+1,j}, \qquad g_{i-\frac{1}{2},j} \approx g_{i-1,j},$$

$$g_{i,j+\frac{1}{2}} \approx g_{i,j+1}, \qquad g_{i,j-\frac{1}{2}} \approx g_{i,j-1}.$$

By this numerical approximation, the above approximation takes the following form (comprehensive details are given in [44]):

$$
\begin{aligned}
(div\,(g \cdot \nabla I))_{i,j} \quad &\approx \quad g_{i+1,j}(I_{i+1,j} - I_{i,j}) - g_{i-1,j}(I_{i,j} - I_{i-1,j}) \\
&\quad + g_{i,j+1}(I_{i,j+1} - I_{i,j}) - g_{i,j-1}(I_{i,j} - I_{i,j-1}) \; . \qquad (2.10)
\end{aligned}
$$

An explicit finite difference scheme can be applied on equation (2.10)

$$
\begin{aligned}
I^{n+1}_{i,j} \quad &= \quad I^{n}_{i,j} + \Delta t \; [g_{i+1,j}(I_{i+1,j} - I_{i,j}) - g_{i-1,j}(I_{i,j} - I_{i-1,j}) \\
&\quad + g_{i,j+1}(I_{i,j+1} - I_{i,j}) - g_{i,j-1}(I_{i,j} - I_{i,j-1})] \; . \qquad (2.11)
\end{aligned}
$$

For a stable evolution of the PM model, the time step Δt must be $\Delta t \leq 0.25$.

2.2 Experimental Results

In the experiments, since it has a more stable character, the second choice of the function g, proposed by Perona and Malik [44], is used:

$$
g\,(\nabla I) = -\frac{1}{\sqrt{1 + \frac{|\nabla I|^2}{k^2}}} \; .
$$

In order to choice k, the method given in [53] is used, and in the experiments, 2-norm of the image is used (i.e., $p = 2$). To analyze the performance of the PM model, we calculate SNR (signal-to-noise ratio) improvements known as the simplest and widely used measure of image quality. SNR improvements of the diffusion results have been examined to evaluate the algorithm and they are given for each image.

Figure 2.1 illustrates the results of the classical PM nonlinear process for 30, 50 and 100 iterations by using the time step size of 0.1. SNR of the noisy image and SNR of the final image given in Figure 2.1 (d) have been calculated and an SNR improvement of 5.0765 dB has been achieved.

In Figure 2.2, the PM nonlinear diffusion process is shown for 50, 75 and 100 iterations. For the image displayed in Figure 2.2 (d), an SNR improvement of 6.1469 dB has been obtained.

The PM model is applied to a medical image by using time step size of 0.2 for 60, 150 and 200 iterations and the results are demonstrated in Figure 2.3. During the diffusion process, noise is removed and edges are kept sharp but some isolated noisy points have remained. For the medical image, the nonlinear diffusion process showed an SNR improvement of 6.1458 dB.

Figure 2.4 display the nonlinear diffusion results of Lena image by using the PM model, the time step size of 0.1 for 50, 60 and 75 iterations. An SNR improvement has been calculated as 7.4653 dB for the image given in Figure 2.4 (d).

The edge detection results of the images shown in Figure 2.4 are given in Figure 2.5. As seen from the figure, the edges given in Figure 2.5 (d) are better than that of noisy image given in Figure 2.5 (a). As a result, well-located edges can be obtained after nonlinear diffusion process.

Figure 2.6 shows the comparison of the PM diffusion results for decreasing and fixed k. For decreasing k, the value of k is calculated automatically in each iteration, an SNR improvement of 6.5828 dB has been achieved. For fixed $k = 0.5$, the result is given in Figure 2.6 (c), an SNR improvement of 3.6395 dB has been attained. After many trials, a suitable value of k is obtained and the result is shown in Figure 2.6 (d) for $k = 2$, and an SNR improvement of 6.3847 dB has been achieved. As seen from SNR improvements, better results are obtained for decreasing k. To find a suitable k for each image is a difficult task; thus, using the decreasing k given in [53] is more meaningful for nonlinear diffusion.

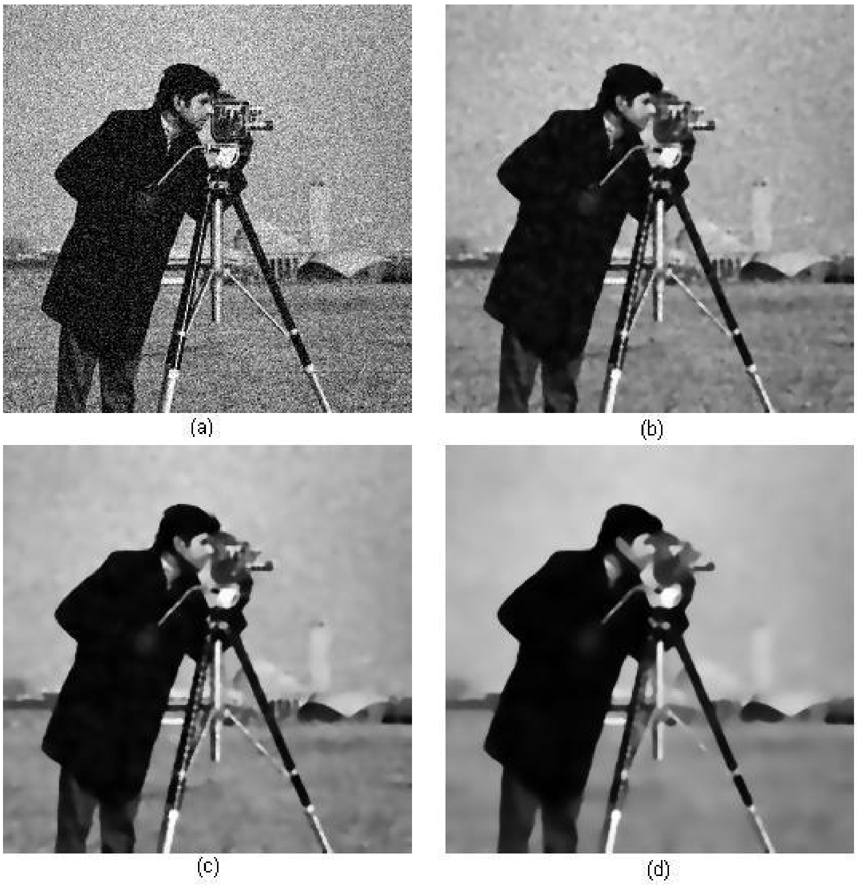

Figure 2.1: (a) The initial noisy image. (b) The result of the PM nonlinear diffusion for 30 iterations with k given by (2.4), (c) the result for 50 iterations and (d) the result for 100 iterations.

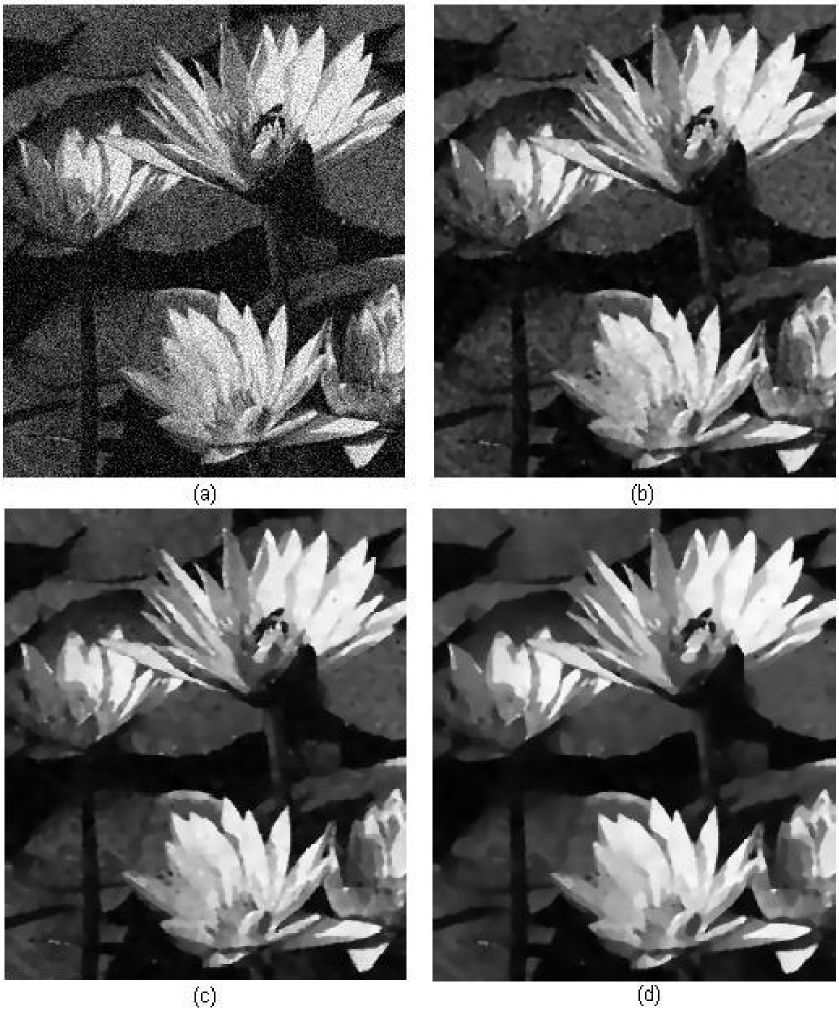

Figure 2.2: Diffusion process by the PM nonlinear diffusion (a) Noisy image. (b) The result of the PM model for 50 iterations, (c) the result for 75 iterations, (d) the result for 100 iterations.

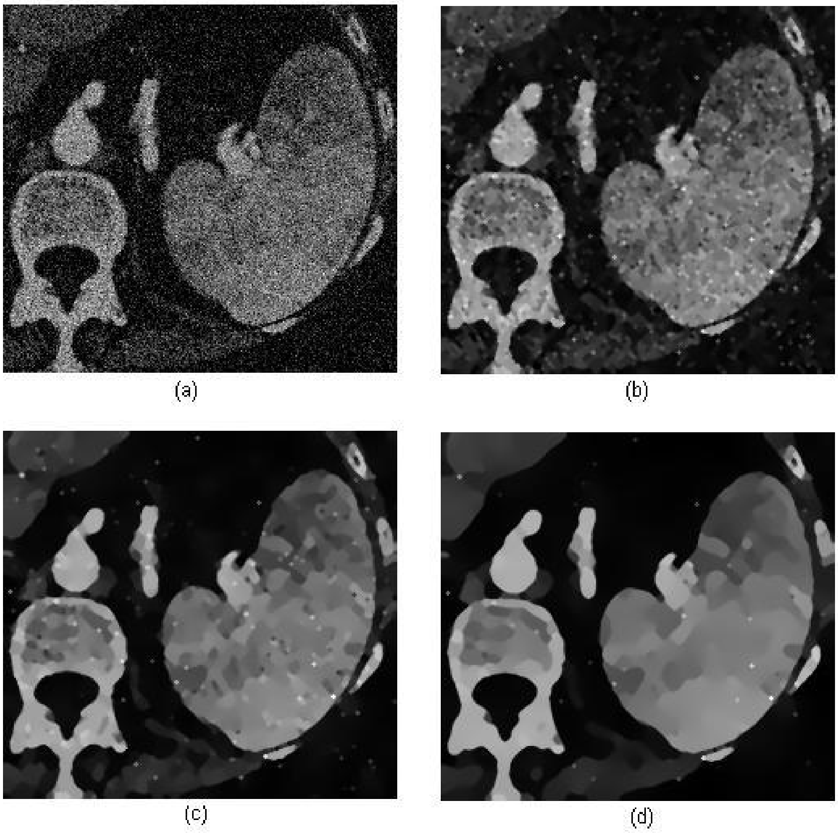

Figure 2.3: Smoothing results of a medical image. (a) The initial noisy image. (b) The result of the PM model for 60 iterations, (c) the result for 150 iterations, (d) the result for 200 iterations.

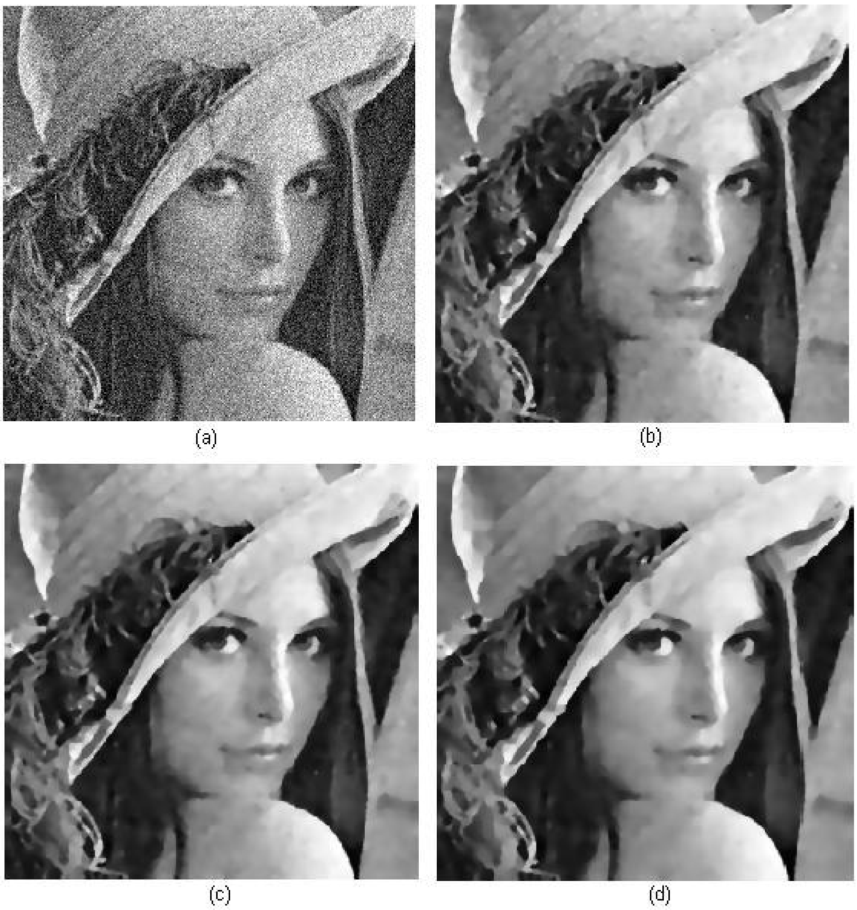

Figure 2.4: (a) Original noisy image. (b) The result of the PM model for 50 iterations, (c) the result for 60 iterations, (d) the result for 75 iterations.

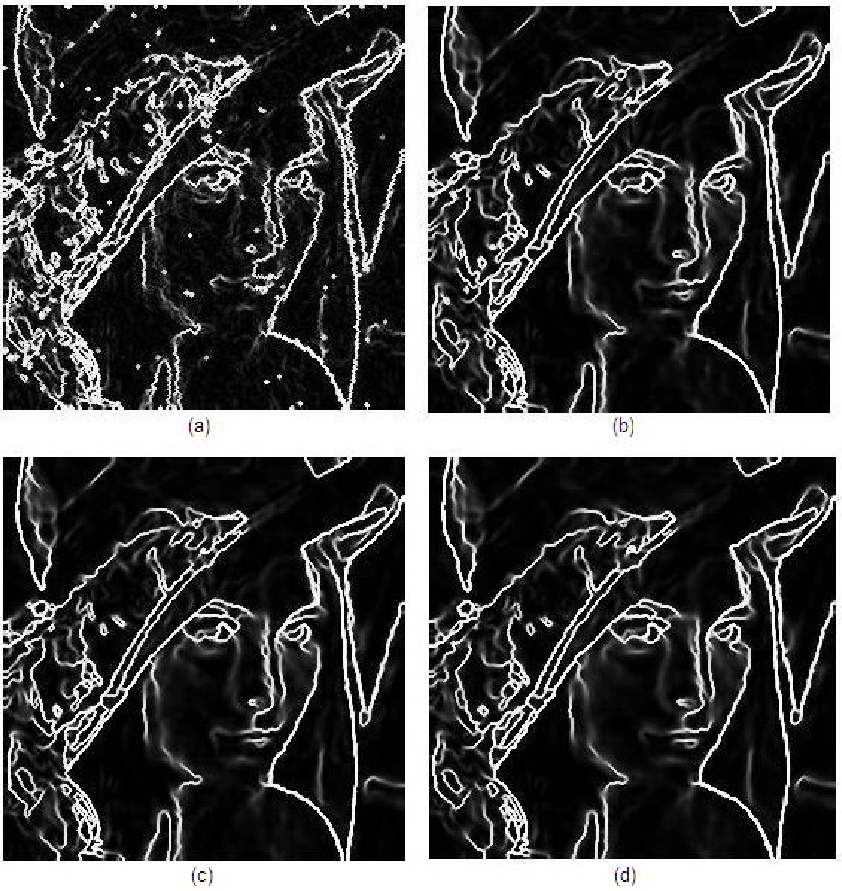

Figure 2.5: Edge detection results applied on the images obtained by nonlinear diffusion given in Figure 2.4.

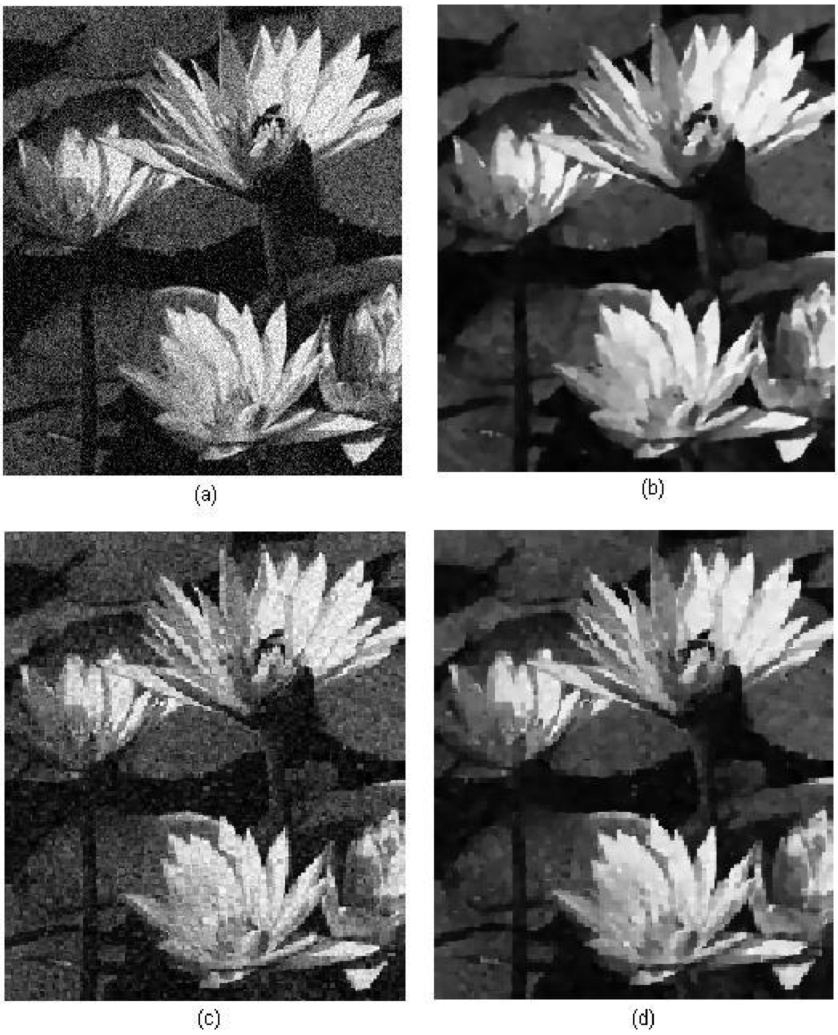

Figure 2.6: Comparasion of fixed k with decreasing k given in [53]. (a) Noisy image. (b) Nonlinear diffusion results by using the PM model for decreasing k, after 75 iterations, (c) for a fixed k=0.5 after 75 iterations, (d) for a constant k=2 after 75 iterations.

CHAPTER 3

MUMFORD AND SHAH APPROACH

Segmentation and denoising tasks are necessary if one desires to find the details of a noisy image. In order to deal with the problem, variational and PDE-based image segmentation and smoothing approaches have been developed and used in many applications such as medical imaging, satellite imaging, tracking, robot vision. Mumford and Shah have introduced a mathematical model that enables both image segmentation and denoising [41]. This model is one of the most widely used models in image segmentation and denoising schemes. The model is a way of finding a partition of an image by using a piecewise smooth representation of the image. The key idea depends on minimizing an energy functional, known as the Mumford-Shah functional, in order to obtain homogeneous representation of the image and detect the boundaries of the smooth regions. They defined a general energy functional E_{MS} which measures the degree of match between an image and its segmentation:

$$E_{MS}(u, C) = \alpha \int\limits_{\Omega - C} |\nabla u|^2 \mathrm{d}x + \beta \int\limits_{\Omega} (u - I)^2 \mathrm{d}x + \mu.\mathrm{length}(C)\ , \tag{3.1}$$

$\Omega \subset \Re^2$ is connected, bounded, open subset representing image domain and it can be a combination of various open subsets;

$$\Omega = \Omega_1 \cup \Omega_2 \cup ... \cup \Omega_n\ .$$

Here, I is the original image defined in the image domain Ω, $C \subset \Omega$ is the set of edge curves, u is a piecewise smooth approximation of I and it is a differentiable function in the open subsets of Ω, which can be discontinuous across the boundary curves C. Furthermore, $\alpha,\ \beta,\ \mu$ are the positive real constants which arrannge the weights of the terms and settle the scale of the segmentation and smoothing.

The first term in $E_{MS}(u, C)$ is the smoothing term, also known as the stabilizing cost, provides u to stay smooth within each region. The second term in $E_{MS}(u, C)$ is the data fidelity term, also known as the deviation cost, measures the least squares distance to the given image. It forces the segmentation to be similar to the original image. The third term in $E_{MS}(u, C)$ is the punishing term, also known as the edge cost, measures the length of the edges and punishes overmuch arc length. It can be defined as

$$\text{length}(C) = \int_C \mathrm{d}s \; .$$

To accomplish a succesful segmentation task, the image function I should be homogenous and should vary smoothly within each Ω_i's. The segmentation problem is defined as of finding a decomposition Ω_i of Ω and an optimal piecewise smooth approximation u of I. This piecewise smooth approximation function u is constructed as the combination of subfunctions u_i which are differentiable in Ω_i.

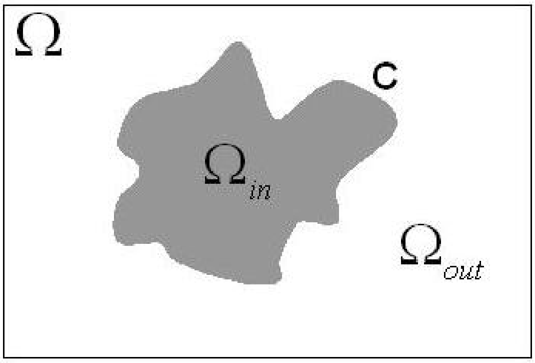

Figure 3.1: The simple image composed of two homogenous regions.

Segmentation of a given image is performed by minimizing the energy functional given in equation (3.1). However, minimizing the functional is not easy due to its complex structure. In particular, in the functional there are two unknowns; the segmented image u and the edge set C. Hence the minimization is done by using the calculus of variations. Firstly, C is fixed and the functional E_{MS} is minimized with respect to u. In the second step, u is fixed and the functional is minimized with respect to C. The constants are fixed during the minimization. For the first step, φ is considered the same type of function as u and the directional derivative of E_{MS} in the direction φ is obtained as follows

$$D_u E_{MS}(u,C)[\varphi] = \lim_{\epsilon \to 0} \frac{E_{MS}(u+\epsilon\varphi, C) - E_{MS}(u,C)}{\epsilon} \ ,$$

$$\begin{aligned}
E_{MS}(u+\epsilon\varphi, C) - E_{MS}(u,C) &= \alpha \int\limits_{\Omega - C} |\nabla(u+\epsilon\varphi)|^2 \mathrm{d}x + \beta \int\limits_{\Omega} (u+\epsilon\varphi - I)^2 \mathrm{d}x + \mu . \int\limits_{C} \mathrm{d}s \\
&\quad - \alpha \int\limits_{\Omega - C} |\nabla u|^2 \mathrm{d}x - \beta \int\limits_{\Omega} (u-I)^2 \mathrm{d}x - \mu . \int\limits_{C} \mathrm{d}s \\
&= 2\alpha\epsilon \int\limits_{\Omega - C} \nabla u \cdot \nabla\varphi \mathrm{d}x + \alpha\epsilon^2 \int\limits_{\Omega - C} |\nabla\varphi|^2 \, \mathrm{d}x \\
&\quad + 2\beta\epsilon \int\limits_{\Omega} (u-I)\varphi \mathrm{d}x + \beta\epsilon^2 \int\limits_{\Omega} \varphi^2 \mathrm{d}x \ ,
\end{aligned}$$

$$\begin{aligned}
\lim_{\epsilon \to 0} \frac{E_{MS}(u+\epsilon\varphi, C) - E_{MS}(u,C)}{\epsilon} &= \lim_{\epsilon \to 0} \frac{1}{\epsilon} (2\alpha\epsilon \int\limits_{\Omega - C} \nabla u \cdot \nabla\varphi \mathrm{d}x + \alpha\epsilon^2 \int\limits_{\Omega - C} |\nabla\varphi|^2 \, \mathrm{d}x \\
&\quad + 2\beta\epsilon \int\limits_{\Omega} (u-I)\varphi \mathrm{d}x + \beta\epsilon^2 \int\limits_{\Omega} \varphi^2 \mathrm{d}x) \ ,
\end{aligned}$$

$$\begin{aligned}
\lim_{\epsilon \to 0} (2\alpha \int\limits_{\Omega - C} \nabla u \cdot \nabla\varphi \mathrm{d}x &+ \alpha\epsilon \int\limits_{\Omega - C} |\nabla\varphi|^2 \, \mathrm{d}x \\
&+ 2\beta \int\limits_{\Omega} (u-I)\varphi \mathrm{d}x + \beta\epsilon \int\limits_{\Omega} \varphi^2 \mathrm{d}x) \ ,
\end{aligned}$$

$$D_u E_{MS}(u,C)[\varphi] = 2 \left(\alpha \int\limits_{\Omega - C} \nabla u \cdot \nabla\varphi \mathrm{d}x + \beta \int\limits_{\Omega} (u-I)\varphi \mathrm{d}x \right) . \tag{3.2}$$

Integrating by parts and using the Green's theorem, we obtain

$$\begin{aligned}
\frac{1}{2} D_u E_{MS}(u,C)[\varphi] &= \beta \int\limits_{\Omega - C} (u-I)\varphi \mathrm{d}x - \alpha \int\limits_{\Omega - C} \nabla^2 u \varphi \mathrm{d}x + \alpha \int\limits_{C} \varphi \frac{\partial u}{\partial n} \mathrm{d}s \\
&= \int\limits_{\Omega - C} \varphi \left(\beta(u-I) - \alpha \nabla^2 u \right) \mathrm{d}x + \alpha \int\limits_{C} \varphi \frac{\partial u}{\partial n} \mathrm{d}s \ ,
\end{aligned}$$

where n is the outward unit normal vector of curve C. φ is nonzero in $\Omega - C$ and zero elsewhere and so u satisfies the damped Poisson equation with Neumann boundary condition

$$\nabla^2 u = \frac{\beta}{\alpha}(u - I) \quad \text{in} \quad \Omega - C \,, \tag{3.3}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on} \quad C \,. \tag{3.4}$$

For the sake of simplicity, refering to the Figure 3.1, consider a simple image composed of two homogenous image regions. The above equation becomes the following pair of diffusion equations

$$\nabla^2 u_{in} = \frac{\beta}{\alpha}(u_{in} - I), \quad \frac{\partial u_{in}}{\partial n} = 0 \,, \tag{3.5}$$

$$\nabla^2 u_{out} = \frac{\beta}{\alpha}(u_{out} - I), \quad \frac{\partial u_{out}}{\partial n} = 0 \,, \tag{3.6}$$

where u_{in} and u_{out} represents the value inside and outside the curve C, respectively.

For the second step, the energy functional for the simple image given in Figure 3.1 can be written as

$$\begin{aligned} E(C) &= \alpha\left(\int\limits_{in(C)} |\nabla u_{in}|^2 \, \mathrm{d}x + \int\limits_{out(C)} |\nabla u_{out}|^2 \, \mathrm{d}x \right) \\ &\quad + \beta\left(\int\limits_{in(C)} (u_{in} - I)^2 \mathrm{d}x + \int\limits_{out(C)} (u_{out} - I)^2 \mathrm{d}x \right) + \mu . \int\limits_{C} \mathrm{d}s \,. \end{aligned}$$

Curve evolution process is obtained from the minimization of the energy functional $E(C)$ according to the variation of C with respect to time t. So the curve evolution equation can be written as

$$\frac{\partial C}{\partial t} = \alpha\left(|\nabla u_{in}|^2 - |\nabla u_{out}|^2\right) n + \beta\left((u_{in} - I)^2 - (u_{out} - I)^2\right) n + \mu . \kappa n \,, \tag{3.7}$$

where n is the inward unit normal vector of the curve and κ is the curvature of the curve.

The regional smooth approximations of image function I, obtained from (3.5) and (3.6), u_{in} and u_{out} are used as inputs in the curve evolution equation (3.7). Thus this segmentation algorithm based on diffusion and curve evolution is defined in an iterative manner.

Unfortunately numerical procedures for minimizing the Mumford-Shah functional have some difficulties because the problem is nonconvex and it has numerous local minima, so approximate approaches are proposed. Two such methods are the Ambrosio-Tortorelli approximation and the Chan-Vese active contour method. Ambrosio and Tortorelli developed a practical numerical implementation of the Mumford-Shah model and it is based on an eliptic approximation of the original functional. The Chan-Vese model is a piecewise constant generalization of the Mumford-Shah functional and it is based on level set formulation. The Ambrosio-Tortorelli approximation and the Chan-Vese model are examined in detail in Chapters 4 and 5, respectively.

CHAPTER 4

AMBROSIO-TORTORELLI MODEL

In the Mumford-Shah functional, the unknown edge set (or discontinuity set) makes the minimization difficult because discretization of it is very complicated. A common way is to approximate E_{MS} by a sequence E_ρ of regular functionals defined on Sobolev spaces. As ρ goes to zero, E_ρ converges to E_{MS} in the sense of Γ-convergence (see Appendix) [4]. The edge set C should not be a variable in E_ρ to propose an approximation which is numerically easy to deal with, and so different approaches have been proposed on this problem. The one which has first appeared and widely used is the approximation by elliptic functionals offered by Ambrosio and Tortorelli [3]. They have introduced a smooth edge indicator function v which depends on a parameter ρ. It supposed that on the edges v goes to one and on the smooth regions v goes to zero as ρ goes to zero.

$$v(x) \approx \begin{cases} 0 & \text{if } x \in C \ , \\ 1 & \text{otherwise} \ . \end{cases}$$

The goal of this approximation is to find a sequence of functions (u_ρ, v_ρ) that converges to $(u, 1)$ and so $E_\rho = E_{AT}$ converges to E_{MS}.

The edge term length(C) is replaced by the term $\frac{1}{2}\left(\rho|\nabla v|^2 + \frac{(1-v)^2}{\rho}\right)$. So the new model (AT model) is:

$$E_{AT}(u,v) = \int_\Omega \left(\alpha(v^2|\nabla u|^2) + \beta(u-I)^2 + \frac{1}{2}\left(\rho|\nabla v|^2 + \frac{(1-v)^2}{\rho}\right)\right) \mathrm{d}x \ . \tag{4.1}$$

This approximation describes comprehensive line process regularization where the regularizer has a supplementary constraint introduced by the term $\rho|\nabla v|^2$.

4.1 Euler-Lagrange Equations

In order to minimize the Ambrosio-Tortorelli functional given by equation (4.1), we should determine its gradient and equate it to zero. Using the calculus of variations, the Gateaux derivatives of functional at u and v are taken; so the Euler-Lagrange equations for the functional (4.1) are found.

The Gateaux derivative of E_{AT} at u in the direction φ

$$D_u E_{AT}(u,v)[\varphi] = \lim_{\epsilon\to 0} \frac{E_{AT}(u+\epsilon\varphi, v) - E_{AT}(u,v)}{\epsilon}\ , \tag{4.2}$$

$$\begin{aligned} E_{AT}(u+\epsilon\varphi, v) - E_{AT}(u,v) &= \int_{\Omega} \Big((\alpha v^2|\nabla u + \epsilon\nabla\varphi|^2) + \beta(u+\epsilon\varphi - I)^2\Big)\,\mathrm{d}x \\ &\quad - \int_{\Omega} \Big(\alpha(v^2|\nabla u|^2) + \beta(u - I)^2\Big)\,\mathrm{d}x \\ &= \int_{\Omega} \alpha v^2 \Big(|\nabla u + \epsilon\nabla\varphi|^2 - |\nabla u|^2\Big)\,\mathrm{d}x \\ &\quad + \int_{\Omega} \beta\Big((u+\epsilon\varphi - I)^2 - (u-I)^2\Big)\,\mathrm{d}x\ , \end{aligned} \tag{4.3}$$

- $$\begin{aligned} |\nabla u + \epsilon\nabla\varphi|^2 - |\nabla u|^2 &= |\nabla u|^2 + 2\epsilon\nabla u\cdot\nabla\varphi + \epsilon^2|\nabla\varphi|^2 - |\nabla u|^2 \\ &= 2\epsilon\nabla u\cdot\nabla\varphi + \epsilon^2|\nabla\varphi|^2\ , \end{aligned} \tag{4.4}$$

- $$\begin{aligned} (u+\epsilon\varphi - I)^2 - (u-I)^2 &= (u-I)^2 + 2(u-I)\epsilon\varphi + \epsilon^2\varphi^2 - (u-I)^2 \\ &= 2\epsilon(u-I)\varphi + \epsilon^2\varphi^2\ . \end{aligned} \tag{4.5}$$

Substituting the above results (4.4) and (4.5) in equation (4.3), we obtain

$$E_{AT}(u+\epsilon\varphi,v)-E_{AT}(u,v)=\int\limits_{\Omega}\left(\alpha v^2\left(2\epsilon\nabla u\cdot\nabla\varphi+\epsilon^2|\nabla\varphi|^2\right)+\beta\left(2\epsilon(u-I)\varphi+\epsilon^2\varphi^2\right)\right)\mathrm{d}x\ , \tag{4.6}$$

$$\lim_{\epsilon\to 0}\frac{E_{AT}(u+\epsilon\varphi,v)-E_{AT}(u,v)}{\epsilon}=\int\limits_{\Omega}\left(2\alpha v^2\nabla u\cdot\nabla\varphi+2\beta(u-I)\varphi\right)\mathrm{d}x\ . \tag{4.7}$$

If Green's theorem is applied to the first term, the following equation is obtained.

- $$\int\limits_{\Omega}v^2\nabla u\cdot\nabla\varphi\mathrm{d}x=-\int\limits_{\Omega}\varphi\nabla\cdot(v^2\nabla u)\mathrm{d}x+\int\limits_{\partial\Omega}v^2\frac{\partial u}{\partial n}\varphi\mathrm{d}s\ , \tag{4.8}$$

$$D_uE_{AT}(u,v)[\varphi]=-\int\limits_{\Omega}2\alpha\varphi\nabla\cdot(v^2\nabla u)\mathrm{d}x+\int\limits_{\partial\Omega}2\alpha v^2\frac{\partial u}{\partial n}\varphi\mathrm{d}s+\int\limits_{\Omega}2\beta(u-I)\varphi\mathrm{d}x\ . \tag{4.9}$$

Here, $\frac{\partial u}{\partial n}=0$ and $D_uE_{AT}(u,v)[\varphi]=0$ for all admissable φ which are 0 on boundary $\partial\Omega$.

$$D_uE_{AT}(u,v)[\varphi]=\int\limits_{\Omega}\left(2\beta(u-I)-2\alpha\nabla\cdot(v^2\nabla u)\right)\varphi\mathrm{d}x=0\ . \tag{4.10}$$

From the fundamental lemma of the calculus of variations

$$2\beta(u-I)-2\alpha\nabla\cdot(v^2\nabla u)=0\ . \tag{4.11}$$

Dividing both sides by -2α, we obtain

$$\nabla\cdot(v^2\nabla u)-\frac{\beta}{\alpha}(u-I)=0\ .$$

The Gateaux derivative of E_{AT} at v in the direction ψ is given by

$$D_vE_{AT}(u,v)[\psi]=\lim_{\epsilon\to 0}\frac{E_{AT}(u,v+\epsilon\psi)-E_{AT}(u,v)}{\epsilon}\ , \tag{4.12}$$

$$\begin{aligned}
E_{AT}(u, v + \epsilon\psi) - E_{AT}(u, v) \quad = \quad & \int\limits_{\Omega} \left((\alpha(v + \epsilon\psi)^2|\nabla u|^2) + \frac{1}{2}\left(\rho|\nabla v + \epsilon\nabla\psi|^2 + \frac{(1 - v - \epsilon\psi)^2}{\rho}\right)\right) \mathrm{d}x \\
& - \int\limits_{\Omega} \left(\alpha(v^2|\nabla u|^2) + \frac{1}{2}\left(\rho|\nabla v|^2 + \frac{(1 - v)^2}{\rho}\right)\right) \mathrm{d}x \\
= \quad & \int\limits_{\Omega} \alpha\left((v + \epsilon\psi)^2 - v^2\right)|\nabla u|^2 \mathrm{d}x + \int\limits_{\Omega} \frac{\rho}{2}\left(|\nabla v + \epsilon\nabla\psi|^2 - |\nabla v|^2\right) \mathrm{d}x \\
& - \int\limits_{\Omega} \frac{1}{2\rho}\left((1 - v - \epsilon\psi)^2(1 - v)^2\right) \mathrm{d}x \,, \qquad (4.13)
\end{aligned}$$

- $$\begin{aligned}
(v + \epsilon\psi)^2 - v^2 \quad &= \quad v^2 + 2v\epsilon\psi + \epsilon^2\psi^2 - v^2 \\
&= \quad 2v\epsilon\psi + \epsilon^2\psi^2 \,, \qquad (4.14)
\end{aligned}$$

- $$\begin{aligned}
|\nabla v + \epsilon\nabla\psi|^2 - |\nabla v|^2 \quad &= \quad |\nabla v|^2 + 2\epsilon\nabla v \cdot \nabla\psi + \epsilon^2|\nabla\psi|^2 - |\nabla v|^2 \\
&= \quad 2\epsilon\nabla v \cdot \nabla\psi + \epsilon^2|\nabla\psi|^2 \,, \qquad (4.15)
\end{aligned}$$

- $$\begin{aligned}
(1 - v - \epsilon\psi)^2 - (1 - v)^2 \quad &= \quad (1 - v)^2 + 2(1 - v)\epsilon\psi + \epsilon^2\psi^2 - (1 - v)^2 \\
&= \quad -2(1 - v)\epsilon\psi + \epsilon^2\psi^2 \,. \qquad (4.16)
\end{aligned}$$

We substitute the above results (4.14), (4.15) and (4.16) in equation (4.13):

$$\begin{aligned}
E_{AT}(u, v + \epsilon\psi) - E_{AT}(u, v) \quad = \quad & \int\limits_{\Omega} \alpha\left(2v\epsilon\psi + \epsilon^2\psi^2\right)|\nabla u|^2 \mathrm{d}x \int\limits_{\Omega} \frac{\rho}{2}\left(2\epsilon\nabla v \cdot \nabla\psi + \epsilon^2|\nabla\psi|^2\right) \mathrm{d}x \\
& + \int\limits_{\Omega} \frac{1}{2\rho}\left(-2(1 - v)\epsilon\psi + \epsilon^2\psi^2\right) \mathrm{d}x \,, \qquad (4.17)
\end{aligned}$$

$$\lim_{\epsilon \to 0} \frac{E_{AT}(u, v + \epsilon\psi) - E_{AT}(u, v)}{\epsilon} = \int\limits_{\Omega} 2\alpha v\psi|\nabla u|^2 \mathrm{d}x + \int\limits_{\Omega} \rho\nabla v \cdot \nabla\psi \mathrm{d}x - \int\limits_{\Omega} \frac{1 - v}{\rho}\psi \mathrm{d}x \,. \quad (4.18)$$

Applying the Green's theorem to the second term, we obtain:

$$\int_{\Omega} \nabla v \cdot \nabla \psi \mathrm{d}x = -\int_{\Omega} \nabla^2 v \psi \mathrm{d}x + \int_{\partial\Omega} \frac{\partial v}{\partial n} \psi \mathrm{d}s \ . \tag{4.19}$$

$$D_v E_{AT}(u,v)[\psi] = \int_{\Omega} 2\alpha v \psi |\nabla u|^2 \mathrm{d}x - \int_{\Omega} \rho \nabla^2 v \psi \mathrm{d}x + \int_{\partial\Omega} \rho \frac{\partial v}{\partial n} \psi \mathrm{d}s - \int_{\Omega} \frac{1-v}{\rho} \psi \mathrm{d}x \ . \tag{4.20}$$

Here, $\frac{\partial v}{\partial n} = 0$ and $D_v E_{AT}(u,v)[\psi] = 0$ for all admissable ψ which are 0 on boundary $\partial\Omega$.

$$\int_{\Omega} \left(2\alpha v |\nabla u|^2 - \rho \nabla^2 v - \frac{1-v}{\rho} \right) \psi \mathrm{d}x = 0 \ . \tag{4.21}$$

By using the fundamental lemma of the calculus of variations, we obtain:

$$2\alpha v |\nabla u|^2 - \rho \nabla^2 v - \frac{1-v}{\rho} = 0 \ . \tag{4.22}$$

Both sides can be divided by $-\rho$

$$\nabla^2 v + \frac{1-v}{\rho^2} - \frac{2\alpha v |\nabla u|^2}{\rho} = 0 \ .$$

Thus, the Euler-Lagrange equations are obtained:

$$\nabla \cdot (v^2 \nabla u) - \frac{\beta}{\alpha}(u - I) = 0, \qquad \frac{\partial u}{\partial n}|_{\partial\Omega} = 0 \ , \tag{4.23}$$

$$\nabla^2 v + \frac{1-v}{\rho^2} - \frac{2\alpha v |\nabla u|^2}{\rho} = 0, \qquad \frac{\partial v}{\partial n}|_{\partial\Omega} = 0 \ . \tag{4.24}$$

4.2 Numerical Approximations

In order to determine the solutions of Euler-Lagrange equations (4.23) and (4.24) approximately in the distributed sense, we construct the following formulas to update u and v.

$$\frac{\partial u}{\partial t} = \nabla.(v^2 \nabla u) - \frac{\beta}{\alpha}(u - I) \, , \tag{4.25}$$

$$\frac{\partial v}{\partial t} = \nabla^2 v - \frac{2\alpha v |\nabla u|^2}{\rho} - \frac{v - 1}{\rho^2} \, . \tag{4.26}$$

Indeed, these are diffusion equations and each of them minimizes a convex quadratic functional. In particular, keeping v fixed, the first equation minimizes

$$\int_{\Omega} \left(\alpha v^2 |\nabla u|^2 + \beta (u - I)^2 \right) \mathrm{d}x \, . \tag{4.27}$$

Similarly, keeping u fixed the second equation minimizes

$$\int_{\Omega} \left(\rho |\nabla v|^2 + \frac{1 + 2\alpha\rho |\nabla u|^2}{\rho} \left(\frac{1}{1 + 2\alpha\rho |\nabla u|^2} - v \right)^2 \right) \mathrm{d}x \, . \tag{4.28}$$

The Euler-Lagarange equations (4.23) and (4.24) or their diffusion equations equivalent (4.25) and (4.26) have in general multiple solutions. So any numerical solution technique is strongly dependent on the initial choices of the functions u and v. A reasonable selection as the initial state of u, $u(x, 0)$ is the original image itself. Then a suitable initial state for v can be found as by letting $t \to \infty$ in equation (4.26)

$$\nabla^2 v - \frac{2\alpha v |\nabla u|^2}{\rho} - \frac{v - 1}{\rho^2} = 0 \, ,$$

v can be extracted from above equation as

$$v = \frac{1 + \rho^2 \nabla^2 v}{1 + 2\alpha\rho |\nabla u|^2} \, . \tag{4.29}$$

Since ρ is small enough, $\rho^2 \approx 0$. Thus the initial state of the edge indicator function v can be written as

$$v(x, 0) \approx \frac{1}{1 + 2\alpha\rho |\nabla u|^2} = \frac{1}{1 + 2\alpha\rho |\nabla I|^2} \, . \tag{4.30}$$

A final note about equations (4.25) and (4.26) is that even though they are a coupled set of nonlinear initial value problems for the pair u and v, interestingly, they are linear equations; (4.25) is linear with respect to u, when v is fixed and conversely (4.26) is linear with respect to v, when u is fixed.

$$\frac{\partial u}{\partial t} = \nabla.(v^2 \nabla u) - \frac{\beta}{\alpha}(u - I) \,,$$

$$\frac{\partial v}{\partial t} = \nabla^2 v - \frac{2\alpha v|\nabla u|^2}{\rho} - \frac{v-1}{\rho^2} \,.$$

Method of finite difference approximation is used in order to discretize and obtain an approximate solution of these equations. Regularization terms on the right hand side of each equations (the first terms) are evaluated at time t. Bias terms in the right hand side of each equations are evaluated at time t+1.

$$\frac{d}{\mathrm{d}t} u_{i,j}(t) = \nabla.\left(v_{i,j}^2(t)\nabla u_{i,j}(t)\right) - \frac{\beta}{\alpha}(u_{i,j}(t) - I_{i,j}) \,, \tag{4.31}$$

$$\frac{d}{\mathrm{d}t} v_{i,j}(t) = \nabla^2 v_{i,j}(t) - \frac{2\alpha v_{i,j}(t)|\nabla u_{i,j}(t)|^2}{\rho} - \frac{v_{i,j}(t) - 1}{\rho^2} \,. \tag{4.32}$$

h is the space step, Δt is the time step, $d_{i,j}(t) = v_{i,j}^2(t)$ and

$$\nabla u_{i,j} = \begin{pmatrix} \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{h} \\ \\ \frac{u_{i,j+\frac{1}{2}} - u_{i,j-\frac{1}{2}}}{h} \end{pmatrix} ,$$

$$(d\nabla u)_{i,j} = \begin{pmatrix} \frac{d_{i,j}u_{i+\frac{1}{2},j} - d_{i,j}u_{i-\frac{1}{2},j}}{h} \\ \\ \frac{d_{i,j}u_{i,j+\frac{1}{2}} - d_{i,j}u_{i,j-\frac{1}{2}}}{h} \end{pmatrix} ,$$

$$\begin{aligned} (\nabla\,(d\nabla u))_{i,j} \quad = \quad & \frac{d_{i+\frac{1}{2},j}u_{i+1,j} - d_{i-\frac{1}{2},j}u_{i,j}}{h^2} - \frac{d_{i+\frac{1}{2},j}u_{i,j} - d_{i-\frac{1}{2},j}u_{i-1,j}}{h^2} \\ & + \frac{d_{i,j+\frac{1}{2}}u_{i,j+1} - d_{i,j-\frac{1}{2}}u_{i,j}}{h^2} - \frac{d_{i,j+\frac{1}{2}}u_{i,j} - d_{i,j-\frac{1}{2}}u_{i,j-1}}{h^2} \,, \end{aligned}$$

$$\begin{aligned}(\nabla\,(d\nabla u))_{i,j} = \frac{1}{h^2}\,&(d_{i+\frac{1}{2},j}(u_{i+1,j}-u_{i,j}) - d_{i-\frac{1}{2},j}(u_{i,j}-u_{i-1,j})\\ &+d_{i,j+\frac{1}{2}}(u_{i,j+1}-u_{i,j}) - d_{i,j-\frac{1}{2}}(u_{i,j}-u_{i,j-1}))\ ,\end{aligned}$$

$$\begin{aligned}\Delta_+^x u_{i,j} &= u_{i+1,j}-u_{i,j}\ , & \Delta_-^x u_{i,j} &= u_{i,j}-u_{i-1,j}\ ,\\ \Delta_+^y u_{i,j} &= u_{i,j+1}-u_{i,j}\ , & \Delta_-^y u_{i,j} &= u_{i,j}-u_{i,j-1}\ ,\\ d_{i\pm\frac{1}{2},j} &= \tfrac{d_{i\pm1,j}+d_{i,j}}{2}\ , & d_{i,j\pm\frac{1}{2}} &= \tfrac{d_{i,j}+d_{i,j\pm1}}{2}\ ,\end{aligned}$$

$$\begin{aligned}(\nabla\,(d\nabla u))_{i,j} &= \frac{1}{h^2}\,(d_{i+\frac{1}{2},j}\Delta_+^x u_{i,j} - d_{i-\frac{1}{2},j}\Delta_-^x u_{i,j}\\ &\quad + d_{i,j+\frac{1}{2}}\Delta_+^y u_{i,j} - d_{i,j-\frac{1}{2}}\Delta_-^y u_{i,j})\ . \qquad (4.33)\end{aligned}$$

We use the above equation (4.33) in equation (4.31) to obtain

$$\begin{aligned}\frac{u_{i,j}^{t+1}-u_{i,j}^t}{\Delta t} &= \frac{1}{h^2}\,(d_{i+\frac{1}{2},j}^t\Delta_+^x u_{i,j}^t - d_{i-\frac{1}{2},j}^t\Delta_-^x u_{i,j}^t + d_{i,j+\frac{1}{2}}^t\Delta_+^y u_{i,j}^t - d_{i,j-\frac{1}{2}}^t\Delta_-^y u_{i,j}^t)\\ &\quad - \frac{\beta}{\alpha}\,(u_{i,j}^{t+1}-I_{i,j})\ ,\end{aligned}$$

$$\begin{aligned}u_{i,j}^{t+1} &= \frac{1}{1+\frac{\Delta t\beta}{\alpha}}(u_{i,j}^t + \frac{\Delta t\beta}{\alpha}I_{i,j} + \frac{\Delta t}{h^2}(d_{i+\frac{1}{2},j}^t\Delta_+^x u_{i,j}^t - d_{i-\frac{1}{2},j}^t\Delta_-^x u_{i,j}^t\\ &\quad + d_{i,j+\frac{1}{2}}^t\Delta_+^y u_{i,j}^t - d_{i,j-\frac{1}{2}}^t\Delta_-^y u_{i,j}^t))\ . \qquad (4.34)\end{aligned}$$

$$\nabla v_{i,j} = \begin{pmatrix}\frac{v_{i+\frac{1}{2},j}-v_{i-\frac{1}{2},j}}{h}\\ \\ \frac{v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{h}\end{pmatrix}\ ,$$

$$\begin{aligned}(\nabla^2 v)_{i,j} &= \frac{v_{i+1,j}-v_{i,j}}{h^2} - \frac{v_{i,j}-v_{i-1,j}}{h^2} + \frac{v_{i,j+1}-v_{i,j}}{h^2} - \frac{v_{i,j}-v_{i,j-1}}{h^2}\\ &= \frac{1}{h^2}(\Delta_+^x v_{i,j} - \Delta_-^x v_{i,j} + \Delta_+^y v_{i,j} - \Delta_-^y v_{i,j})\ . \qquad (4.35)\end{aligned}$$

$$\nabla u_{i,j} = \begin{pmatrix} \frac{u_{i+1,j}-u_{i-1,j}}{2h} \\ \\ \frac{u_{i,j+1}-u_{i,j-1}}{2h} \end{pmatrix} ,$$

$$\begin{aligned} |\nabla u|^2_{i,j} &= (\frac{u_{i+1,j} - u_{i-1,j}}{2h})^2 + (\frac{u_{i,j+1} - u_{i,j-1}}{2h})^2 \\ &= \frac{(u_{i+1,j} - u_{i-1,j})^2 + (u_{i,j+1} - u_{i,j-1})^2}{(2h)^2} , \end{aligned} \tag{4.36}$$

the equations above (4.35) and (4.36) are substituted in equation (4.32) to obtain

$$\begin{aligned} \frac{v^{t+1}_{i,j} - v^t_{i,j}}{\Delta t} &= \frac{1}{h^2}(\Delta^x_+ v^t_{i,j} - \Delta^x_- v^t_{i,j} + \Delta^y_+ v^t_{i,j} - \Delta^y_- v^t_{i,j}) \\ &- \frac{2\alpha}{\rho}\left(\frac{(u_{i+1,j} - u_{i-1,j})^2 + (u_{i,j+1} - u_{i,j-1})^2}{(2h)^2}\right) v^{t+1}_{i,j} - \frac{v^{t+1}_{i,j} - 1}{\rho^2} , \end{aligned}$$

$$v^{t+1}_{i,j} = \left(1 + \Delta t\left(\frac{2\alpha\rho A + 1}{\rho^2}\right)\right)^{-1} \left(v^t_{i,j} + \frac{\Delta t}{\rho^2} + \frac{\Delta t}{h^2}(\Delta^x_+ v^t_{i,j} - \Delta^x_- v^t_{i,j} + \Delta^y_+ v^t_{i,j} - \Delta^y_- v^t_{i,j})\right) , \tag{4.37}$$

where

$$A = \left(\frac{(u_{i+1,j} - u_{i-1,j})^2 + (u_{i,j+1} - u_{i,j-1})^2}{(2h)^2}\right) .$$

$$u^{t+1}_{i,j} = \frac{1}{1 + \frac{\Delta t\beta}{\alpha}}(u^t_{i,j} + \frac{\Delta t\beta}{\alpha} g_{i,j} + \frac{\Delta t}{h^2}(d^t_{i+\frac{1}{2},j}\Delta^x_+ u^t_{i,j} - d^t_{i-\frac{1}{2},j}\Delta^x_- u^t_{i,j} + d^t_{i,j+\frac{1}{2}}\Delta^y_+ u^t_{i,j} - d^t_{i,j-\frac{1}{2}}\Delta^y_- u^t_{i,j})) , \tag{4.38}$$

$$v^{t+1}_{i,j} = \left(1 + \Delta t\left(\frac{2\alpha\rho A + 1}{\rho^2}\right)\right)^{-1} \left(v^t_{i,j} + \frac{\Delta t}{\rho^2} + \frac{\Delta t}{h^2}(\Delta^x_+ v^t_{i,j} - \Delta^x_- v^t_{i,j} + \Delta^y_+ v^t_{i,j} - \Delta^y_- v^t_{i,j})\right) . \tag{4.39}$$

Equations (4.38) and (4.39) are discretized approximated update formulas for the smooth image function u and the edge indicator function v.

4.3 Experimental Results

In this section, the segmentation results of the Ambrosio-Tortorelli (AT) algorithm are demonstrated. In all experiments, the Neumann type boundary conditions are used along the image boundary. Figure 4.1 illustrates segmentation results of the Lena image obtained using the AT algorithm with different scale space parameters and after 100 iterations. Figure 4.1 (a) is the original image of Lena. The segmentation results for $\alpha = 5, \ \beta = 1.5, \ \rho = 0.001$ are given in Figure 4.1 (b)-(c). Figure 4.1 (b) and Figure 4.1 (c) demonstrate the function u and the edge function $1 - v$ respectively. Figure 4.1 (d)-(e) are the reconstruction results for $\alpha = 3, \ \beta = 0.5, \ \rho = 0.001$. Figure 4.2 demonstrates segmentation results for different value of the ρ with $\alpha = 2, \ \beta = 0.1$ and 100 iterations. Figure 4.2 (a)-(b) are the results obtained by using $\rho = 0.01$ and Figure 4.2 (c)-(d) are the results obtained by using $\rho = 0.001$. It can be observed from the results that the smaller value of ρ is, the less detailed construction is obtained. In Figure 4.3, the reconstruction results are demonstrated for decreasing β values and keeping α fixed after 200 iterations.

Figure 4.3 (a) is the original image. Figure 4.3 (b)-(c) are the resonstruction results of the image by using $\alpha = 0.75, \ \beta = 0.01, \ \rho = 0.01$ and Figure 4.3 (d)-(e) are the resonstruction results of the image by changing $\beta = 0.001$. When β is decreased while keeping α fixed, the diffusion is so strong that some details are lost but the noise is not eliminated completely. The same results are obtained in Figure 4.4 for the image of penguens. As seen from the reconstruction results for small value of $\beta = 0.001$ in Figure 4.4 (f)-(g), some details have been lost. If we increase the value of α and keep $\frac{\alpha}{\beta}$ fixed, the penalty of the length term is decreased and we obtain more detailed reconstruction. Figure 4.5 demonstrates the results of the AT with $\rho = 0.001$ and 100 iterations. Figure 4.5 (b)-(c) illustrate segmentation results with $\alpha = 1, \ \beta = 0.01$, (d)-(e) show the results with $\alpha = 2, \ \beta = 0.02$ and finally (f)-(g) show the results with $\alpha = 4, \ \beta = 0.04$. For increasing the value of α and β in the same proportion, more detailed edges can be obtained.

The segmentation results of the venice image corrupted with 10% salt and pepper noise are demonstrated in Figure 4.6. The results given in Figure 4.6 (b)-(c) are obtained by using $\alpha = 1, \ \beta = 0.01, \ \rho = 0.001$, in Figure 4.6 (d)-(e) are obtained obtained by using $\alpha = 1, \ \beta = 0.001$. In this case, strong noise could not be eliminated entirely.

The segmentation results of a texture image are shown in Figure 4.7. Figure 4.7 (b)-(c) illustrate the reconstruction results for $\alpha = 0.5,\ \beta = 0.05,\ \rho = 0.01$ and Figure 4.7 (d)-(e) illustrate the reconstruction results for $\alpha = 15,\ \beta = 1,\ \rho = 0.01$. During the process, the texture regions are also smoothed out, while noise is removed.

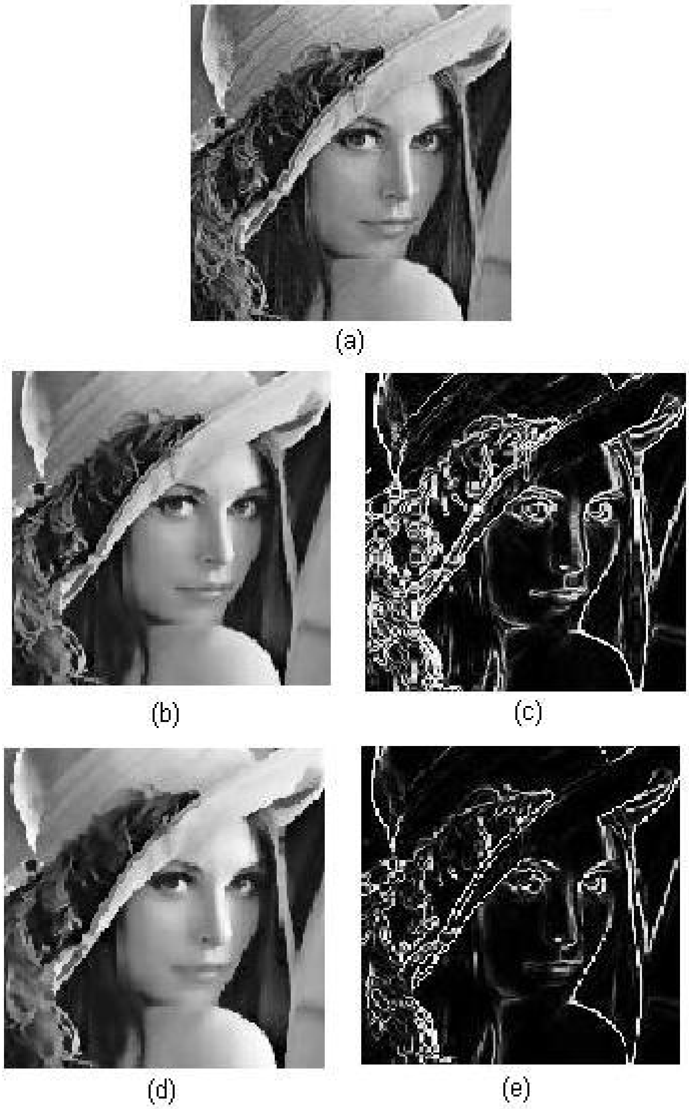

Figure 4.1: Segmentation results of the AT. (a) Original image. (b)-(c) Reconstruction results of the image by using $\alpha = 5, \ \beta = 1.5, \ \rho = 0.001$. (d)-(e) Reconstruction by $\alpha = 3, \ \beta = 0.5, \ \rho = 0.001$.

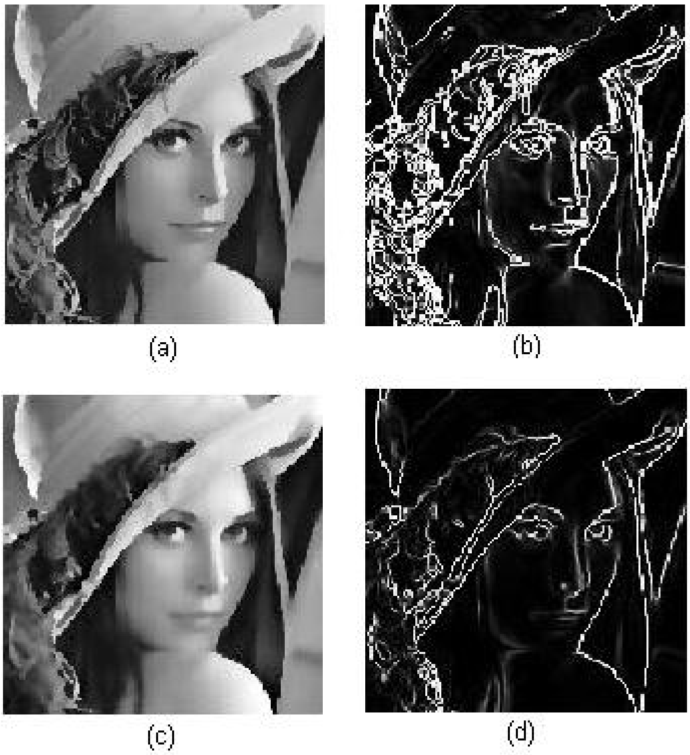

Figure 4.2: Reconstruction results for different value of ρ. (a)-(b) are the results for $\rho = 0.01$, (c)-(d) are the results for $\rho = 0.001$.

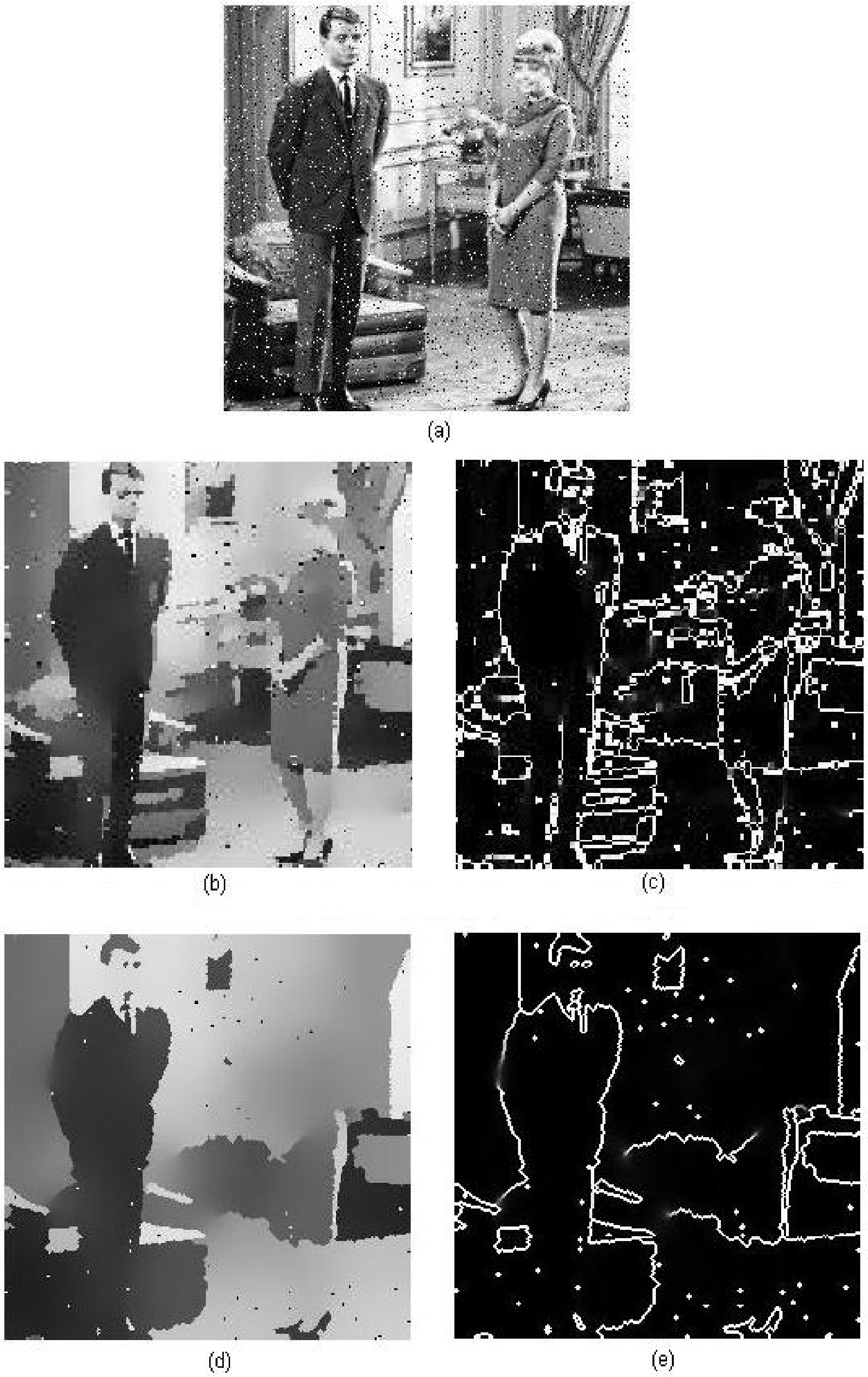

Figure 4.3: Reconstruction results for decreasing β and fixed α.

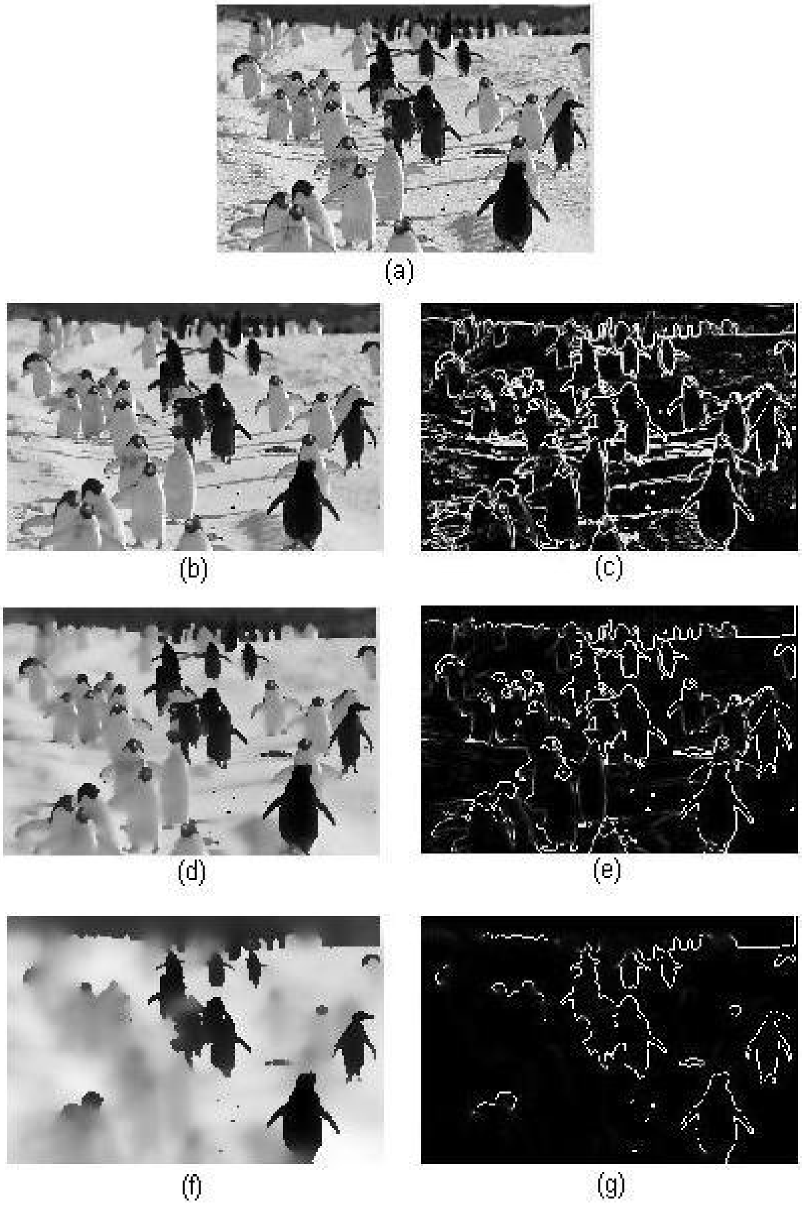

Figure 4.4: Reconstruction results for decreasing β with 100 iterations. (a) is the original image. (b)-(c) reconstruction results for $\alpha = 2$, $\beta = 0.5$, $\rho = 0.001$. (d)-(e) recostruction results for $\beta = 0.01$. (f)-(g) reconstruction results for $\beta = 0.001$.

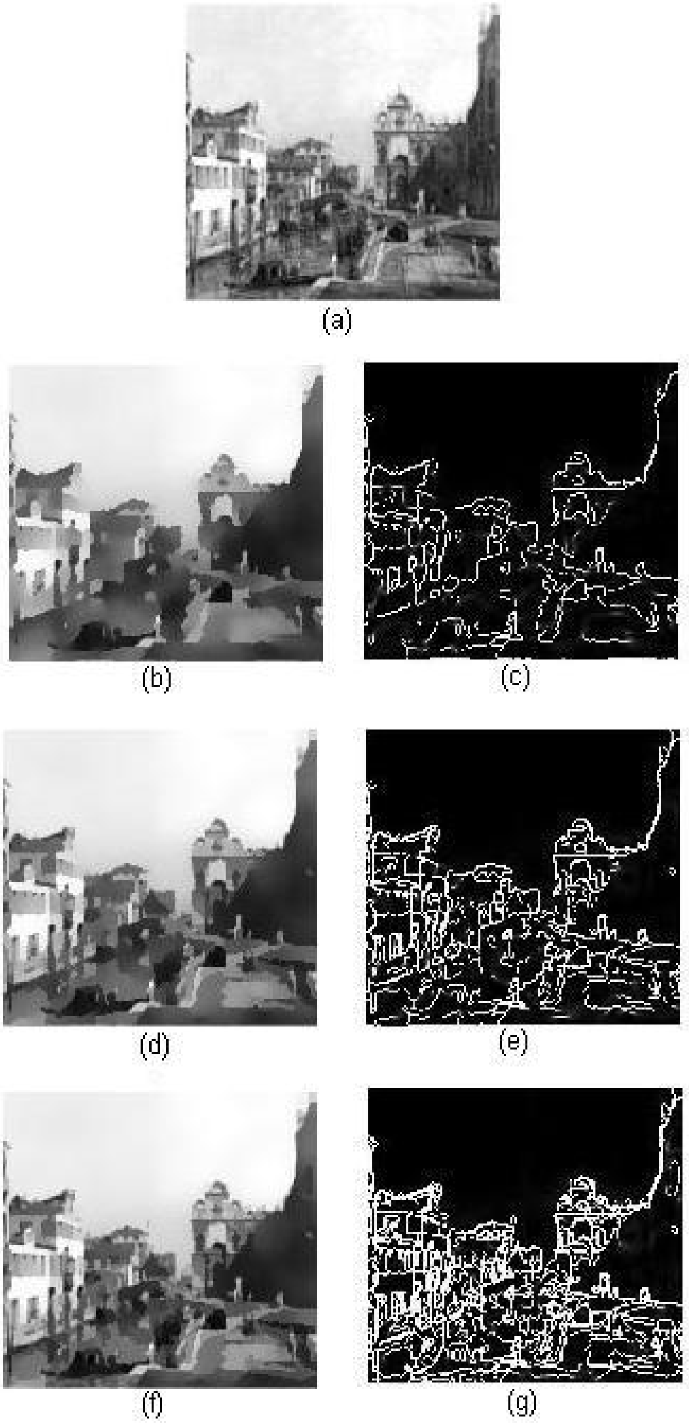

Figure 4.5: Reconstruction results for increasing α and β.

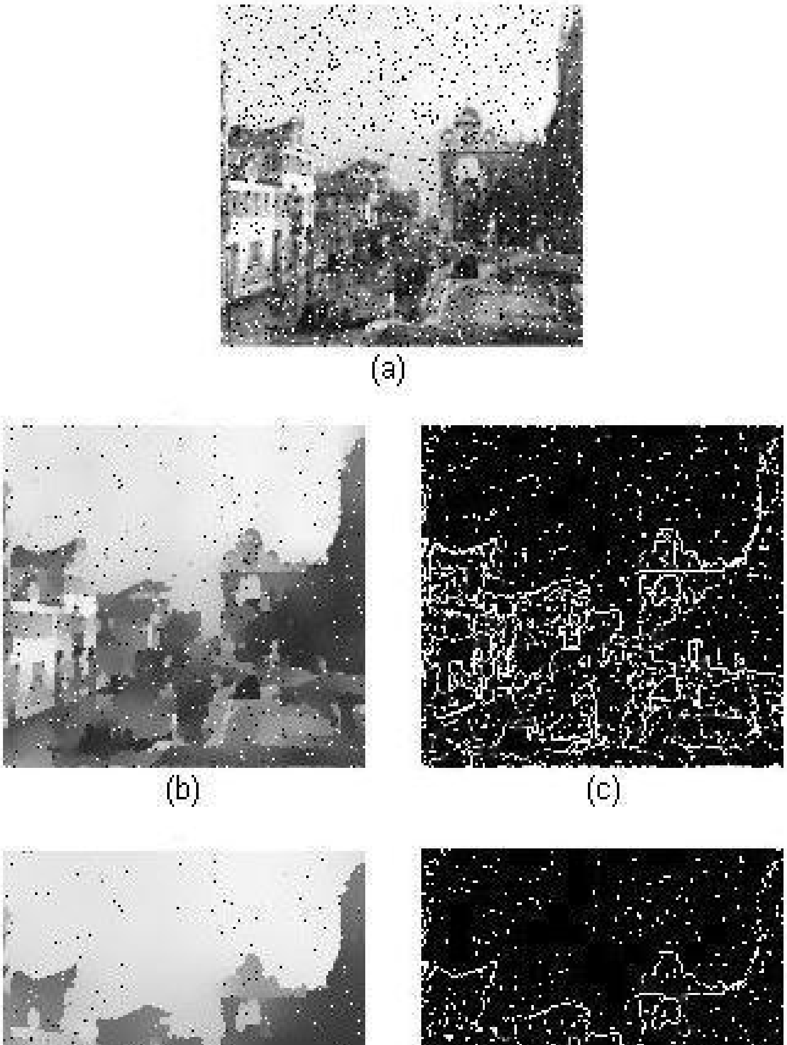

Figure 4.6: Reconstruction results of a very noisy image.

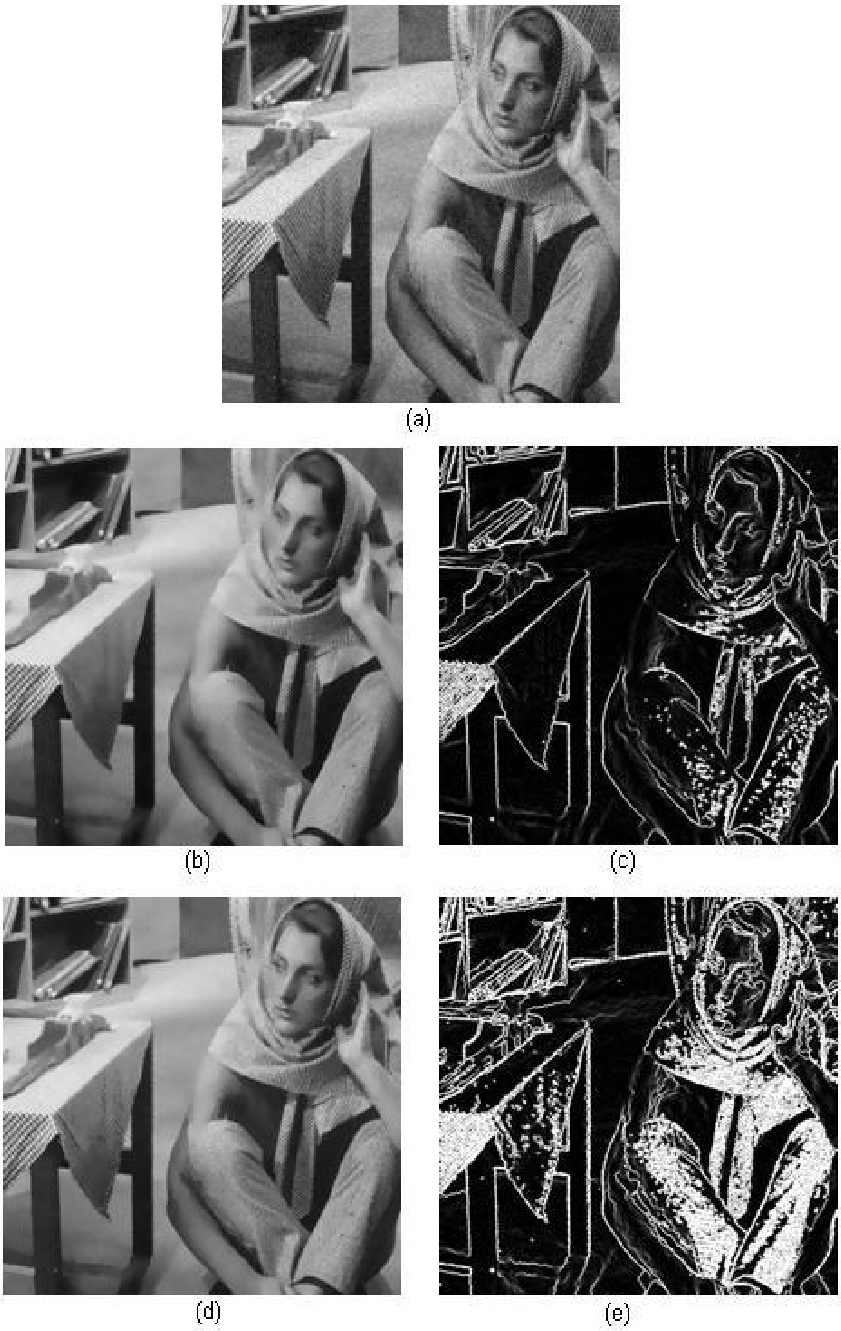

Figure 4.7: Reconstruction results for a texture image.

CHAPTER 5

CHAN-VESE MODEL

In order to detect the edges in an image, Chan and Vese introduced a different approximation of the Mumford-Shah energy functional [9]. This new model is based on curve evolution and level set method introduced by Osher and Sethian [42]. In the literature, it is known as the Chan-Vese active contour model (CV model) which is a region-based segmentation algorithm. It is more robust to noise and less sensitive to the initialization than other active contour techniques.

As oppossed to the classical snake algortihms [6, 32, 57, 59], this model is not based on the gradient of the image to stop the evolving curve on the boundaries of the object, and does not use control points to interpolate the contour C. The interior contours can be detected automatically and the initial curve does not depend on the objects. The main idea of this model is to find the boundaries of the objects via an evolving implicit surface. The zero level set of this surface is a moving curve and this curve detects the object in the image. Owing to this implicit representation, the topological changes such as splitting and merging can be detected automatically. According to this method, an image is separated into homogenous regions by using this level set function. These regions consist of the image parts with close mean values. The energy functional introduced by Chan and Vese [9] is defined as

$$\begin{aligned} E_{CV}(C, c_1, c_2) &= \mu.(\text{length}(C)) + \nu.\text{area}(inside(C)) \\ &\quad + \lambda_1 \int\limits_{inside(C)} |I - c_1|^2 \mathrm{d}x + \lambda_2 \int\limits_{outside(C)} |I - c_2|^2 \mathrm{d}x \,, \end{aligned} \tag{5.1}$$

where I is the given image, $C(s) : [0, 1] \to \Re^2$ is the curve, c_1 and c_2 are the constants and depending on C, $\mu \geq 0$, $\nu \geq 0$, $\lambda_1, \lambda_2 > 0$ are fixed parameters.

5.1 Relation with the MS functional

Chan and Vese intorduced a curve evolution representation of the Mumford-Shah energy functional discussed in Chapter 3.

$$E_{MS}(u,C) = \beta \int\limits_{\Omega} (I-u)^2 \mathrm{d}x + \alpha \int\limits_{\Omega - C} |\nabla u|^2 \mathrm{d}x + \mu.\mathrm{length}(C) \ .$$

Mumford and Shah indicated a reduced form of their model, known as the minimal partition problem. In this case, u is separated into constant functions, i.e., $u = c_i$ and is equal to average of I on each region Ω_i of Ω.

The Chan-Vese model is considered piecewise constant generalization of the minimal partion problem of the Mumford-Shah functional. It can be considered as a particular case of the reduced form of the Mumford-Shah problem for $\nu = 0$, $\lambda_1 = \lambda_2 = \lambda$. In this particular case, the edge C is considered as a snake or an active contour and the approximation u of I is defined as

$$u = \begin{cases} c_1, & \mathrm{inside}(C) \ , \\ c_2, & \mathrm{outside}(C) \ , \end{cases}$$

where c_1 is the average of I inside C and c_2 is the average of I outside C.

The Chan-Vese model is based on the Mumford-Shah functional but it is slightly different. In the Mumford-Shah model the aim is to find an approximation of the initial image composed of homogenous regions. In the Chan-Vese model the idea is to find the boundaries of objects in an image. They are both used for segmenting an image, but which one is more appropriate depends on the kind of the image that we handle. If we deal with an image in which all contours are important and we would like to find recognition of all contours divided by sharp edges, the Mumford-Shah model can be used. If we are interested in an image in which specific objects are significant and we want to find the boundaries of objects, we may use the han-Vase active contour model. Among the images given in Figure 5.1 and 5.2, the Mumford-Shah functional is more appropriate for the images given in the former and the Chan-Vese model can be applied to the images given in the later [4].

Figure 5.1: Images suitable for MS model

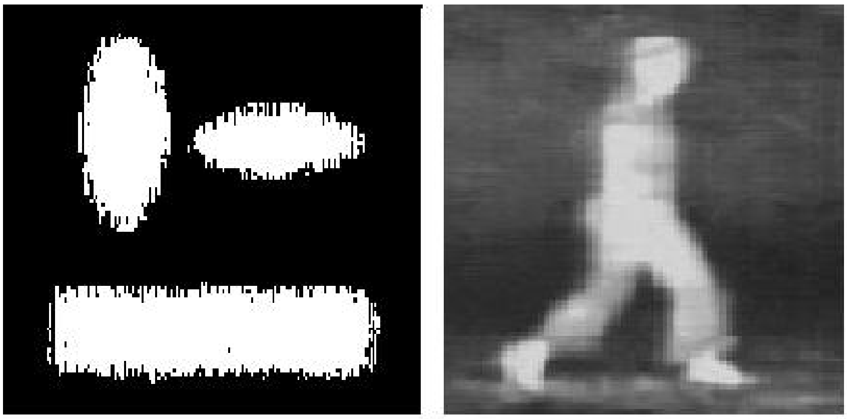

Figure 5.2: Images suitable for CV

5.2 The level set formulation

The level set method is used for minimizing the Chan-Vese energy functional in order to deal with the problems of the classic active contour models. In the level set formulation, there is no explicit contour C and implicit representation is given by the zero level of a Lipschitz function ϕ instead of C. This implicit curve physically overcomes the topological changes, such as splitting and merging. In addition, independent of the initial position, the curve automatically detects interior contours.

The model is reformulated by replacing C with the function ϕ

$$\begin{aligned} E_{CV}(\phi, c_1, c_2) &= \mu.(\text{length}(\phi)) + \nu.\text{area}(\phi) \\ &\quad + \lambda_1 \int\limits_{\phi \geq 0} |I - c_1|^2 \mathrm{d}x + \lambda_2 \int\limits_{\phi < 0} |I - c_2|^2 \mathrm{d}x \,. \end{aligned} \tag{5.2}$$

$\phi : \Re^N \rightarrow \Re$ is defined as [9]

$$\begin{cases} C = \left\{ x \in \Re^N : \phi(x) = 0 \right\} \,, \\ \text{inside}(C) = \left\{ x \in \Re^N : \phi(x) > 0 \right\} \,, \\ \text{outside}(C) = \left\{ x \in \Re^N : \phi(x) < 0 \right\} \,. \end{cases}$$

Heaviside function H and the one dimensional Dirac measure δ are used to express the terms in the energy.

$$H(\phi) = \begin{cases} 1 & \text{if } \phi \geq 0 \,, \\ 0 & \text{if } \phi < 0 \,, \end{cases}$$

$$\delta(\phi) = \frac{d}{d\phi} H(\phi) \,.$$

The terms in the energy functional can be replaced as follows:

$$\text{length}(\phi) = \int\limits_{\Omega} |\nabla H(\phi)| \mathrm{d}x = \int\limits_{\Omega} \delta(\phi) |\nabla \phi| \mathrm{d}x \,,$$

$$\text{area}(\phi) = \int\limits_{\Omega} H(\phi) \mathrm{d}x \,,$$

$$\int\limits_{\phi \geq 0} |I - c_1|^2 \mathrm{d}x = \int\limits_{\Omega} |I - c_1|^2 H(\phi) \mathrm{d}x \,,$$

$$\int\limits_{\phi < 0} |I - c_2|^2 \mathrm{d}x = \int\limits_{\Omega} |I - c_2|^2 (1 - H(\phi)) \mathrm{d}x \,.$$

Then the energy can be written as:

$$
\begin{aligned}
E_{CV}(\phi, c_1, c_2) &= \mu \int\limits_{\Omega} |\nabla H(\phi)| + \nu \int\limits_{\Omega} H(\phi) \mathrm{d}x + \lambda_1 \int\limits_{\Omega} |I - c_1|^2 H(\phi) \mathrm{d}x \\
&+ \lambda_2 \int\limits_{\Omega} |I - c_2|^2 (1 - H(\phi)) \mathrm{d}x \; . \qquad (5.3)
\end{aligned}
$$

Since the Heaviside function H is not differentiable at 0, we cannot compute the first variation of the E_{AT} and the solution of minimization problem. To obtain a global minimizer, regularized versions of the Heaviside function H and the dirac measure δ are used. The approximations of H and δ can be defined as following

$$
H_\epsilon(\phi) = \frac{1}{2}\left(1 + \frac{2}{\pi} arctan(\frac{\phi}{\epsilon})\right) \; , \qquad (5.4)
$$

$$
\delta_\epsilon(\phi) = H_\epsilon^{'}(\phi) = \frac{1}{\pi} \frac{\epsilon}{\epsilon^2 + \phi^2} \; . \qquad (5.5)
$$

As $\epsilon \rightarrow 0$, these approximations converge to H and δ, respectively.

5.3 Euler-Lagrange Equation

The associated Euler-Lagrange equation of the energy functional can be computed by using the slightly regularized versions of the functions H and δ defined in equations (5.4) and (5.5). H_ϵ is the regularized version of H, δ_ϵ is the regularized version of δ where $\delta_\epsilon = H_\epsilon^{'}$. Then the regularized version of the energy functional E_{CV} can be written as:

$$
\begin{aligned}
E_\epsilon(\phi, c_1, c_2) &= \mu \int\limits_{\Omega} \delta_\epsilon(\phi) |\nabla \phi| \mathrm{d}x + \nu \int\limits_{\Omega} H_\epsilon(\phi) \mathrm{d}x + \lambda_1 \int\limits_{\Omega} |I - c_1|^2 H_\epsilon(\phi) \mathrm{d}x \\
&+ \lambda_2 \int\limits_{\Omega} |I - c_2|^2 (1 - H_\epsilon(\phi)) \mathrm{d}x \; , \qquad (5.6)
\end{aligned}
$$

where $|\nabla H_\epsilon(\phi)| = \delta_\epsilon(\phi)|\nabla \phi|$.

Keeping c_1 and c_2 fixed, E_ϵ is minimized with respect to ϕ. Gateaux derivative (see Appendix) of E_ϵ in the direction φ is used:

$$D_\phi E_\epsilon(\phi, c_1, c_2)[\varphi] = \lim_{t\to 0} \frac{E_\epsilon(\phi + t\varphi, c_1, c_2) - E_\epsilon(\phi, c_1, c_2)}{t} \ .$$

Gateaux derivative of the term $\mu \int\limits_\Omega \delta_\epsilon(\phi)|\nabla\phi|\mathrm{d}x$ in the direction φ is

$$\lim_{t\to 0} \mu \ \frac{\int\limits_\Omega \delta_\epsilon(\phi + t\varphi)|\nabla(\phi + t\varphi)|\mathrm{d}x - \int\limits_\Omega \delta_\epsilon(\phi)|\nabla\phi|\mathrm{d}x}{t} \ . \tag{5.7}$$

Adding and substracting the term $\delta_\epsilon(\phi)|\nabla(\phi + t\varphi)|$ to numerator, one can obtain

$$\lim_{t\to 0} \mu \int\limits_\Omega \frac{\delta_\epsilon(\phi + t\varphi)|\nabla(\phi + t\varphi)| - \delta_\epsilon(\phi)|\nabla(\phi + t\varphi)| + \delta_\epsilon(\phi)|\nabla(\phi + t\varphi)| - \delta_\epsilon(\phi)|\nabla\phi|}{t} \mathrm{d}x$$

$$= \lim_{t\to 0} \mu \int\limits_\Omega \left(\frac{\delta_\epsilon(\phi + t\varphi) - \delta_\epsilon(\phi)}{t} \right) |\nabla(\phi + t\varphi)|\mathrm{d}x + \lim_{t\to 0} \mu \int\limits_\Omega \delta_\epsilon(\phi) \frac{|\nabla(\phi + t\varphi)| - |\nabla\phi|}{t} \mathrm{d}x \ .$$

- $$\lim_{t\to 0} \int\limits_\Omega \left(\frac{\delta_\epsilon(\phi + t\varphi) - \delta_\epsilon(\phi)}{t} \right) |\nabla(\phi + t\varphi)|\mathrm{d}x = \lim_{t\to 0} \int\limits_\Omega \frac{\delta_\epsilon(\phi) + \frac{\partial\delta_\epsilon(\phi)}{\partial\phi} \cdot t\varphi - \delta_\epsilon(\phi)}{t} |\nabla(\phi + t\varphi)|\mathrm{d}x$$
  $$= \int\limits_\Omega \delta_\epsilon(\phi)^{'} \varphi |\nabla\phi| \mathrm{d}x \ .$$

- $$\lim_{t\to 0} \int\limits_\Omega \delta_\epsilon(\phi) \frac{|\nabla(\phi + t\varphi)| - |\nabla\phi|}{t} \mathrm{d}x = \lim_{t\to 0} \int\limits_\Omega \delta_\epsilon(\phi) \frac{|\nabla(\phi)| + \frac{\partial|\nabla(\phi)|}{\nabla(\phi)} \cdot t\nabla\varphi - |\nabla\phi|}{t} \mathrm{d}x$$
  $$= \int\limits_\Omega \delta_\epsilon(\phi) \frac{\nabla(\phi)}{|\nabla\phi|} \nabla\varphi \mathrm{d}x \ .$$

Substituting the above results in equation (5.7), we obtain:

$$\lim_{t\to 0}\mu\,\frac{\int\limits_{\Omega}\delta_{\epsilon}(\phi+t\varphi)|\nabla(\phi+t\varphi)|\mathrm{d}x-\int\limits_{\Omega}\delta_{\epsilon}(\phi)|\nabla\phi|\mathrm{d}x}{t}$$

$$=\mu\int\limits_{\Omega}\left(\delta_{\epsilon}^{'}(\phi)\varphi|\nabla\phi|+\delta_{\epsilon}(\phi)\frac{\nabla\phi\cdot\nabla\varphi}{|\nabla\phi|}\right)\mathrm{d}x\,. \tag{5.8}$$

Gateaux derivative of the term $\nu\int\limits_{\Omega}H_{\epsilon}(\phi)\mathrm{d}x$ in the direction φ is

$$\begin{aligned}\lim_{t\to 0}\nu\int\limits_{\Omega}\frac{H_{\epsilon}(\phi+t\varphi)-H_{\epsilon}(\phi)}{t}\mathrm{d}x &= \lim_{t\to 0}\nu\int\limits_{\Omega}\frac{H_{\epsilon}(\phi)+c-H_{\epsilon}(\phi)}{t}\mathrm{d}x\\ &= \nu\int\limits_{\Omega}H_{\epsilon}^{'}(\phi)\varphi\mathrm{d}x\\ &= \nu\int\limits_{\Omega}\delta_{\epsilon}(\phi)\varphi\mathrm{d}x\,. \end{aligned}\tag{5.9}$$

Gateaux derivative of the term $\lambda_1\int\limits_{\Omega}|I-c_1|^2H_{\epsilon}(\phi)\mathrm{d}x$ in the direction φ is

$$\begin{aligned}\lim_{t\to 0}\lambda_1\int\limits_{\Omega}|I-c_1|^2\frac{H_{\epsilon}(\phi+t\varphi)-H_{\epsilon}(\phi)}{t}\mathrm{d}x &= \lim_{t\to 0}\lambda_1\int\limits_{\Omega}|I-c_1|^2\frac{H_{\epsilon}(\phi)+\frac{\partial H_{\epsilon}(\phi)}{\partial\phi}\cdot t\varphi-H_{\epsilon}(\phi)}{t}\mathrm{d}x\\ &= \lambda_1\int\limits_{\Omega}|I-c_1|^2H_{\epsilon}^{'}(\phi)\varphi\mathrm{d}x\\ &= \lambda_1\int\limits_{\Omega}|I-c_1|^2\delta_{\epsilon}(\phi)\varphi\mathrm{d}x\,. \end{aligned}\tag{5.10}$$

Gateaux derivative of the term $\lambda_2\int\limits_{\Omega}|I-c_2|^2(1-H_{\epsilon}(\phi))\mathrm{d}x$ in the direction φ is

$$\begin{aligned}\lim_{t\to 0}\lambda_2\int\limits_{\Omega}|I-c_2|^2\frac{(1-H_{\epsilon}(\phi+t\varphi))-(1-H_{\epsilon}(\phi))}{t}\mathrm{d}x &= \lim_{t\to 0}\lambda_2\int\limits_{\Omega}|I-c_2|^2\frac{-H_{\epsilon}(\phi)-\frac{\partial H_{\epsilon}(\phi)}{\partial\phi}\cdot t\varphi+H_{\epsilon}(\phi)}{t}\mathrm{d}x\\ &= -\lambda_2\int\limits_{\Omega}|I-c_2|^2H_{\epsilon}^{'}(\phi)\varphi\mathrm{d}x\\ &= -\lambda_2\int\limits_{\Omega}|I-c_2|^2\delta_{\epsilon}(\phi)\varphi\mathrm{d}x\,. \end{aligned}\tag{5.11}$$

Gathering the above derivatives given by equation (5.8), (5.9), (5.10) and (5.11), we obtain the following equation.

$$\int\limits_{\Omega} \mu \left(\delta_\epsilon^{'}(\phi)|\nabla\phi|\varphi + \delta_\epsilon(\phi)\frac{\nabla\phi\cdot\nabla\varphi}{|\nabla\phi|} \right) \mathrm{d}x + \int\limits_{\Omega} \delta_\epsilon(\phi)(\nu + \lambda_1(I-c_1)^2 - \lambda_2(I-c_2)^2)\varphi \mathrm{d}x = 0 \; . \tag{5.12}$$

When Green's theorem (see Appendix) is applied, we obtain

- $$\int\limits_{\Omega} \delta_\epsilon(\phi)\frac{\nabla\phi\cdot\nabla\varphi}{|\nabla\phi|}\mathrm{d}x = \int\limits_{\partial\Omega} \frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}\varphi \mathrm{d}s - \int\limits_{\Omega} \varphi\nabla\cdot\left(\delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right)\mathrm{d}x \; . \tag{5.13}$$

Substituting equation (5.13) in equation (5.12), we get

$$\begin{aligned} \int\limits_{\Omega} \mu\delta_\epsilon^{'}(\phi)|\nabla\phi|\varphi\mathrm{d}x + \int\limits_{\partial\Omega} \mu\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}\varphi\mathrm{d}s - \int\limits_{\Omega} \mu\nabla\cdot\left(\delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right)\varphi\mathrm{d}x \\ + \int\limits_{\Omega} \delta_\epsilon(\phi)(\nu + \lambda_1(I-c_1)^2 - \lambda_2(I-c_2)^2)\varphi\mathrm{d}x = 0 \; . \end{aligned}$$

Developing the divergence operator

$$\int\limits_{\Omega} \mu\nabla\cdot\left(\delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right)\varphi\mathrm{d}x = \int\limits_{\Omega} \mu\delta_\epsilon^{'}(\phi)|\nabla\phi|\varphi\mathrm{d}x + \int\limits_{\Omega} \mu\delta_\epsilon(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\varphi\mathrm{d}x \; ,$$

we finally obtain

$$\begin{aligned} \int\limits_{\partial\Omega} \mu\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}\varphi\mathrm{d}s \quad &- \quad \int\limits_{\Omega} \mu\delta_\epsilon(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\varphi\mathrm{d}x \\ &+ \quad \int\limits_{\Omega} \delta_\epsilon(\phi)(\nu + \lambda_1(I-c_1)^2 - \lambda_2(I-c_2)^2)\varphi\mathrm{d}x = 0 \; , \end{aligned} \tag{5.14}$$

for all admissable φ. Since $\frac{\partial\phi}{\partial n} = 0$

$$\int\limits_{\Omega} \left(\mu\delta_\epsilon(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \delta_\epsilon(\phi)\left(\nu + \lambda_1(I-c_1)^2 - \lambda_2(I-c_2)^2\right)\right)\varphi\mathrm{d}x = 0 \; . \tag{5.15}$$

By using the fundamental lemma of the calculus of variations (see Appendix), the following Euler Lagrange equation for ϕ is obtained.

$$\delta_\epsilon(\phi)\left[\mu\nabla\left(\frac{\nabla\phi}{|\nabla\phi|}\right)-\nu-\lambda_1(I-c_1)^2+\lambda_2(I-c_2)^2\right]=0\ , \tag{5.16}$$

$$\frac{\partial\phi}{\partial n}=0 \quad \text{on } \partial\Omega\ .$$

5.4 Numerical Approximations

In order to solve the Euler-Lagrange equation (5.16), we use the steepest descent algorithm in the distributed sense introducing an artificial time parameter t. The distributed iteration formula for the steepest descent algorithm is given as

$$\frac{\partial\phi}{\partial t}=\delta_\epsilon(\phi)\left[\mu\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)-\nu-\lambda_1(I-c_1)^2+\lambda_2(I-c_2)^2\right]\ , \tag{5.17}$$

$$\phi(t,x,y)=\phi_0(x,y) \quad \text{in } \ \Omega\ ,$$

$$\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}=0 \quad \text{on } \ \partial\Omega\ ,$$

$$c_1(\phi)=\frac{\int\limits_\Omega I H_\epsilon(\phi)\mathrm{d}x}{\int\limits_\Omega H_\epsilon(\phi)\mathrm{d}x}=\text{average}(I) \quad \text{in } \ \phi>0\ , \tag{5.18}$$

$$c_2(\phi)=\frac{\int\limits_\Omega I(1-H_\epsilon(\phi))\mathrm{d}x}{\int\limits_\Omega (1-H_\epsilon(\phi))\mathrm{d}x}=\text{average}(I) \quad \text{in } \ \phi<0\ . \tag{5.19}$$

Method of finite differences is used to approximate the iteration formula. Let h be the space step, Δt time step and $(x_i, y_i) = (ih, jh)$ be the grid points. Let $\phi^n_{i,j} = \phi(n\Delta t, ih, jh)$ be an approximation of $\phi(t, x, y)$ with $n \geq 0$. The finite differences are:

$$\Delta_+^x \phi_{i,j} = \phi_{i+1,j} - \phi_{i,j} \ , \qquad \Delta_-^x \phi_{i,j} = \phi_{i,j} - \phi_{i-1,j} \ ,$$
$$\Delta_+^y \phi_{i,j} = \phi_{i,j+1} - \phi_{i,j} \ , \qquad \Delta_-^y \phi_{i,j} = \phi_{i,j} - \phi_{i,j-1} \ .$$

Discretization of the equation (5.17) yields

$$\frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta t} = \delta_\epsilon(\phi_{i,j}^n)\left(\mu\kappa(\phi_{i,j}^n) - \nu - \lambda_1(I - c_1)^2 + \lambda_2(I - c_2)^2\right) \ ,$$

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n + \Delta t \delta_\epsilon(\phi_{i,j}^n)\left(\mu\kappa(\phi_{i,j}^n) - \nu - \lambda_1(I - c_1)^2 + \lambda_2(I - c_2)^2\right) \ , \tag{5.20}$$

where

$$c_1(\phi_{i,j}^n) = \frac{\sum I H_\epsilon(\phi_{i,j}^n)}{\sum H_\epsilon(\phi_{i,j}^n)} \ , \qquad c_2(\phi_{i,j}^n) = \frac{\sum I(1 - H_\epsilon(\phi_{i,j}^n))}{\sum (1 - H_\epsilon(\phi_{i,j}^n))} \ ,$$

$$H_\epsilon(\phi_{i,j}^n) = \frac{1}{2}\left(1 + \frac{2}{\pi} arctan(\frac{\phi_{i,j}^n}{\epsilon})\right) \ , \tag{5.21}$$

$$\delta_\epsilon(\phi_{i,j}^n) = \frac{1}{\pi}\frac{\epsilon}{\epsilon^2 + (\phi_{i,j}^n)^2} \ , \tag{5.22}$$

$$\kappa(\phi_{i,j}^n) = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right)_{i,j}^n \ , \tag{5.23}$$

$$\nabla\phi_{i,j} = \begin{pmatrix} \frac{\phi_{i+\frac{1}{2},j} - \phi_{i-\frac{1}{2},j}}{h} \\ \\ \frac{\phi_{i,j+\frac{1}{2}} - \phi_{i,j-\frac{1}{2}}}{h} \end{pmatrix} \ ,$$

$$|\nabla\phi|_{i,j} = \sqrt{\left(\frac{\phi_{i+\frac{1}{2},j} - \phi_{i-\frac{1}{2},j}}{h}\right)^2 + \left(\frac{\phi_{i,j+\frac{1}{2}} - \phi_{i,j-\frac{1}{2}}}{h}\right)^2} \ ,$$

$$k = |\nabla\phi|_{i,j} \ ,$$

$$\left(\frac{\nabla\phi}{|\nabla\phi|}\right)_{i,j} = \frac{1}{k}\begin{pmatrix} \dfrac{\phi_{i+\frac{1}{2},j} - \phi_{i-\frac{1}{2},j}}{h} \\ \\ \dfrac{\phi_{i,j+\frac{1}{2}} - \phi_{i,j-\frac{1}{2}}}{h} \end{pmatrix},$$

$$\left(\nabla\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right)_{i,j} = \frac{1}{k}\left(\frac{\phi_{i+1,j} - \phi_{i,j}}{h^2} - \frac{\phi_{i,j} - \phi_{i-1,j}}{h^2} + \frac{\phi_{i,j+1} - \phi_{i,j}}{h^2} - \frac{\phi_{i,j} - \phi_{i,j-1}}{h^2}\right),$$

$$\kappa(\phi^n_{i,j}) = \frac{1}{kh^2}\left(\Delta^x_+\phi^n_{i,j} - \Delta^x_-\phi^n_{i,j} + \Delta^y_+\phi^n_{i,j} - \Delta^y_-\phi^n_{i,j}\right) . \qquad (5.24)$$

Initial state for ϕ is an arbitrary curve located around the objects in the image.

5.5 Experimental Results

In this section, experimental results of the Chan-Vese active contour models are presented. This model is applied to several synthetic and real images having different kinds of boundary and shapes. In the implementations, the parameters have been chosen as follows: $\mu > 0,\ \nu = 0,\ \lambda_1 = \lambda_2 = 1,\ h = 1$ and $\Delta t = 0.1$. Since the derivatives of the H and δ functions do not exist everywhere, the approximations H_ϵ and δ_ϵ (for $\epsilon = 1$) are used to provide the computation of a global minimiser. The parameter μ is changed in each experiment. For detecting most of the edges of any size, μ should be small, while for detecting larger objects, μ should be large. The implementation of the CV model on a synthetic image is presented in Figure 5.3. Figure 5.3 (a) shows three objects with different shapes and the initial curve, Figure 5.3 (b) shows the contour extraction of these different objects and detection of interior contour. Figure 5.3 (c)-(d) give results of the CV model for the same and noisy image. As seen in the figure, the model works well in a noisy environment and handles the topological changes. In Figure 5.4, the results of the CV model on an image with smooth contours are demonstrated. Figure 5.4 (a) gives original image, silhouette of a man, with initial contour and Figure 5.4 (b) displays the contour extraction of the man. The model successfully finds the smooth contours. Figure 5.5 illustrates detection of four people located in different places. The original real image and the initial contour is given in Figure 5.5 (a), as seen in the figure, the position of the initial contour can be somewhere in the image and it does not necessarily enclose the people. Detection of two people with smooth contours from a real image is presented in Figure 5.6.

The curve is spread and than divided into two pieces which locate the boundaries of people. Figure 5.7 and Figure 5.8 demonstrate the results of the CV algorithm applied to outcomes of the Perona-Malik and the Ambrosio-Tortorelli models, respectively. Figure 5.7 (a) shows the original image of penguins. The result obtained using the PM model in 100 iterations is presented in Figure 5.7 (b). Figure 5.7 (c) shows the result of the PM model with initial contour, Figure 5.7 (d), (e) and (f) present the CV evolutions on the result of the PM model after 300, 500 and 1000 iterations, respectively. Figure 5.8 (a) is the image of penguins given previously in Figure 5.7 (a). Figure 5.8 (b)-(c) present the results of the AT approximation of the MS functional in 300 iterations. Figure 5.8 (d) shows the edges given in Figure 5.8 (c) with initial contour and Figure 5.8 (d) gives the CV evolution after 100 iterations. As seen from the figures, the CV evolution on the edges obtained by using the AT model is very well.

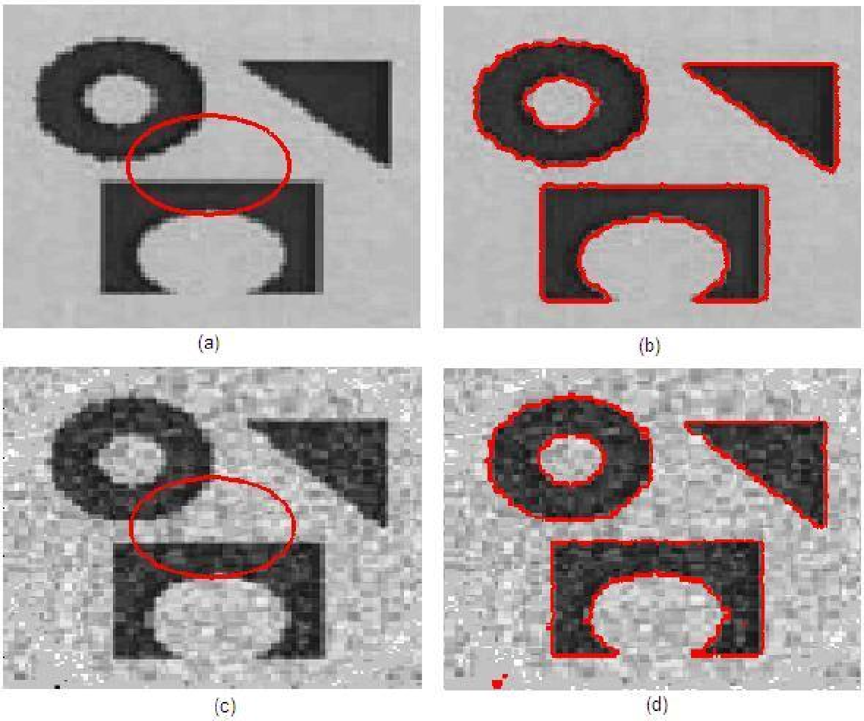

Figure 5.3: (a) Original image with initial contour. (b) The result of the CV algorithm. (c)-(d) Original noisy image with initial contour and result of CV respectively.

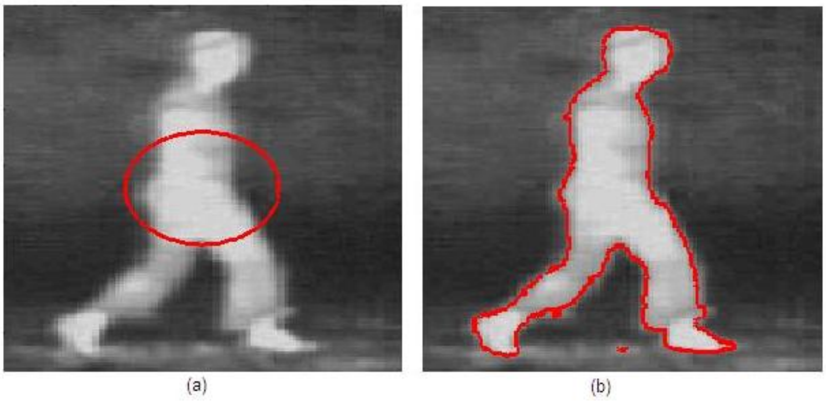

Figure 5.4: (a) Siluet of a man with initial contour. (b) The result of the CV model.

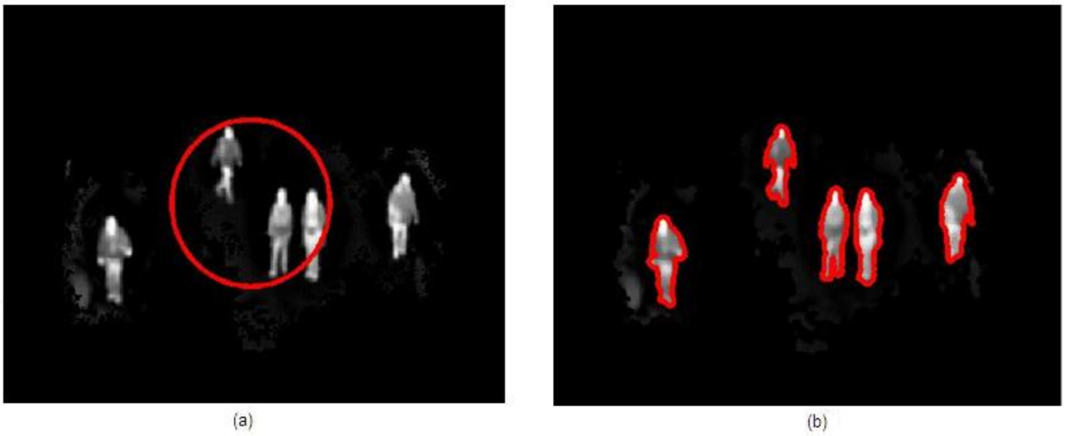

Figure 5.5: (a) Initial image with four people. (b) The result of the CV: four people has been detected succesfully.

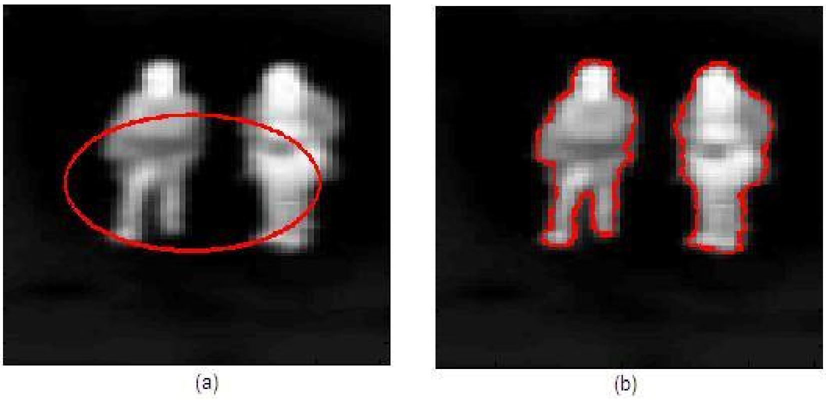

Figure 5.6: Detection of two people with the CV algorithm.

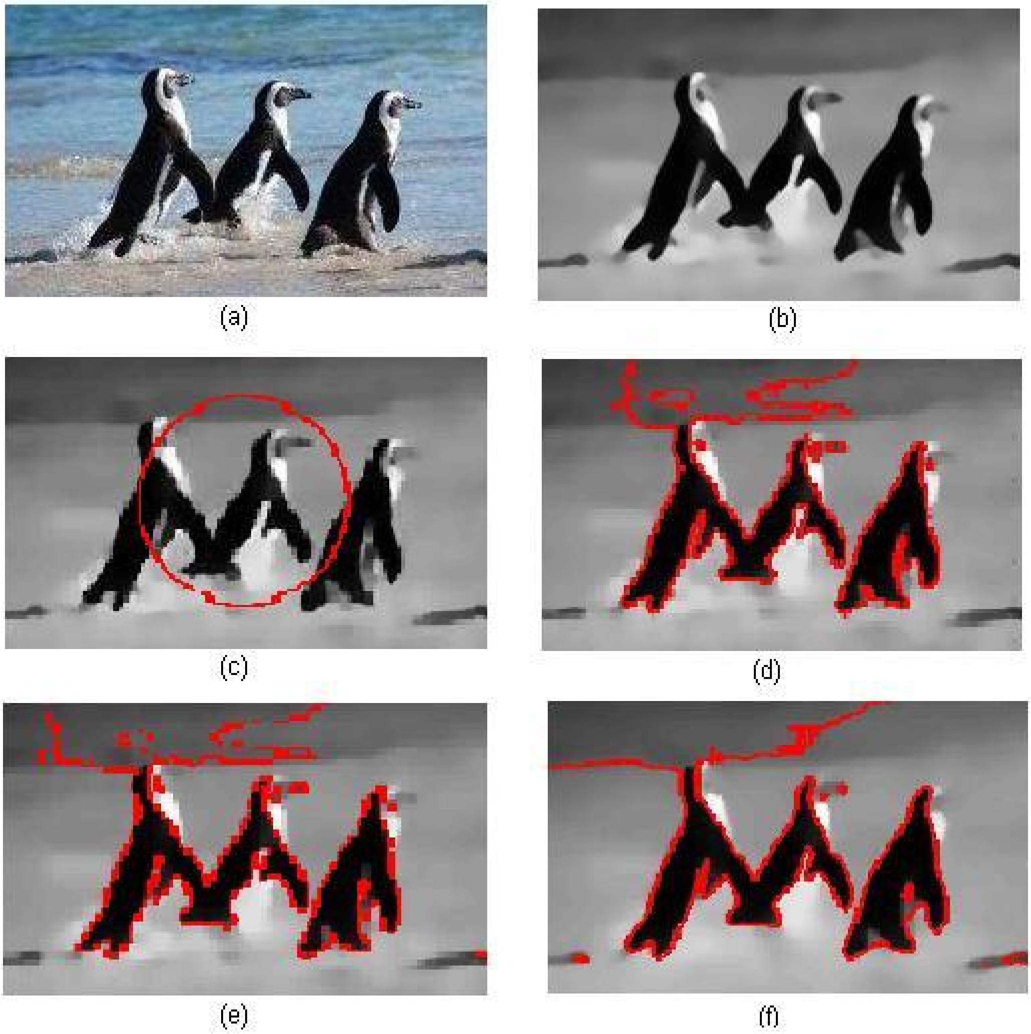

Figure 5.7: (a) Original image, (b) The result of the PM model after 100 iterations, (c) The PM model with initial contour, (d) is the result of the CV model after 300 iteraions, (e) The CV after 500 iterations, (f) The CV after 1000 iterations.

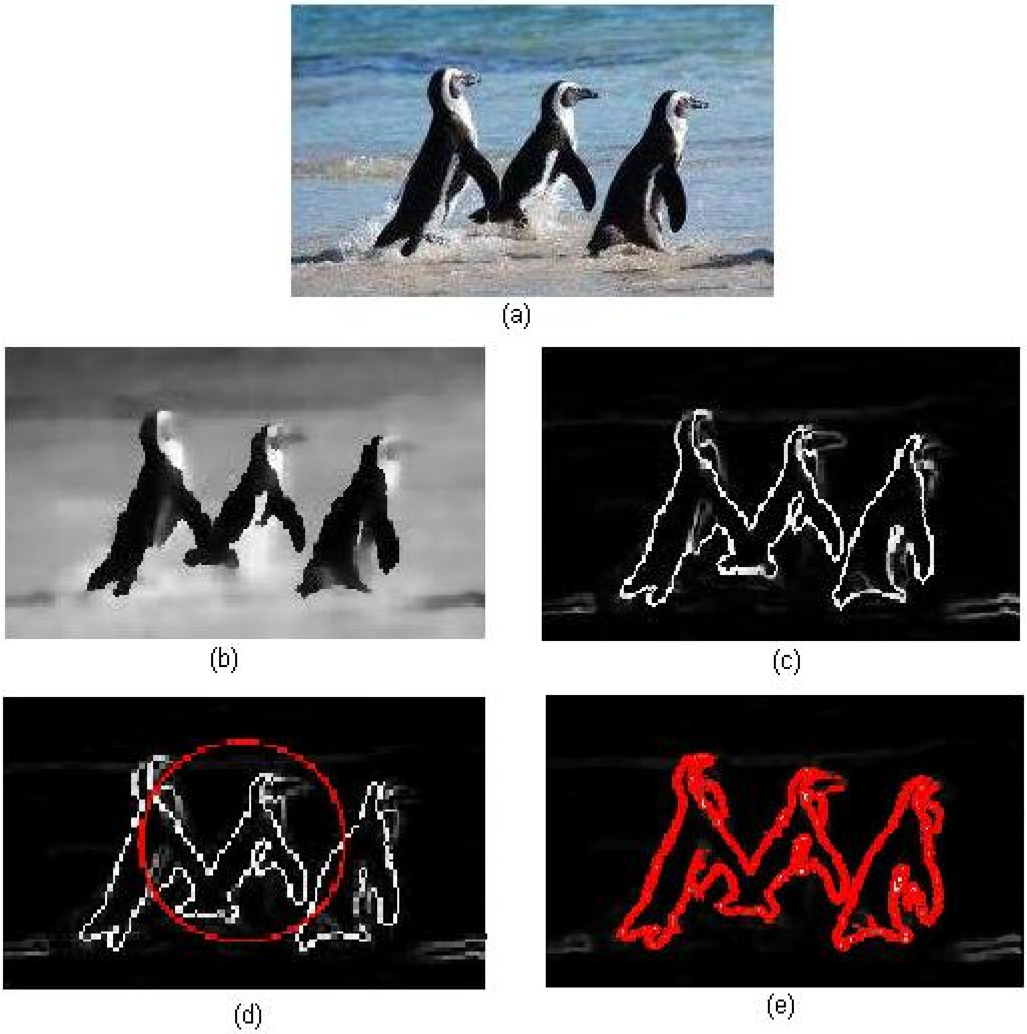

Figure 5.8: (a) Original image. (b)-(c) The results of the AT model after 100 iterations with $\alpha = 1, \beta = 0.1, \rho = 0.001$. (d) Initial contour. (e) The result of the CV model after 300 iterations.

CHAPTER 6

SNAKES: ACTIVE CONTOUR MODELS

The technique of active contours or snakes have been first introduced by Kass et. al. in 1987 [32]. The main idea behind this method is to evolve a curve by means of energy minimization. Since its publication, it has gained popularity and researches have been done to generalize the forms of the contours and to cope with the convergence and stability problems encountered during the energy minimization process. By now, this technique is widely and successfully used in many image segmentation and motion tracking tasks. An active contour is defined as a curve or a surface within an image and it detects the boundary of a desired object. The contour starts from an initial position and moves towards the closest object boundary; this is why they are sometimes called “snakes”. The initial curve location should be determined close enough to the object boundary for successful process. The prior knowledge of the shape of the object may be needed for this purpose. This curve evolves under the influence of image features inside and outside it and stops at the boundary of the object of interest according to a stopping criteria. The edge detector function defined as a function of ∇I, where I is the given image, signifies when the curve is placed at the boundary of an object. The performance of the evolution is dependent on the shape and location of the initial curve.

Mathematically, a snake is defined as

$$C(s) = (x(s), y(s)) \, , \;\; s \in [0, 1] \; .$$

The energy of the snake is defined as

$$E^{*}_{snk}(C) = E_{int}(C) + E_{ext}(C) \; . \tag{6.1}$$

The total energy of the snake is the integral of the energy at each point. To find the best fit between a snake and an object's shape, we minimize the total energy. The energy functional to be minimized is

$$\begin{aligned} E_{snk}(C) &= \int_0^1 E^*_{snk}(C)\mathrm{d}s \\ &= \int_0^1 \left(E_{int}(C) + E_{ext}(C)\right)\mathrm{d}s\ . \end{aligned} \tag{6.2}$$

The energy functional consists of internal and external energy terms. The internal energy keeps the model smooth during the deformation. It prevents curve to deform too much to maintain the curve's original shape. The external energy pulls or pushes the snake toward the edges of the object. It guides the snake by using image information such as intensity, gradient magnitude etc. There is an inverse association between energy and image features: strong features have low energy and weak features have high energy.

Active contours are classified into two types: parametric active contours [31, 32, 57, 59] and geometric active contours [6, 9, 36, 43] according to the choice of the mathematical representations for curves [61]. Parametric active contours are represented explicitly, typically with splines in a Lagrangian formulation. The main idea is to evolve the curve by adjusting the corresponding parameters or control points. With this method, it is difficult to handle the topological changes such as splitting or merging, but its computational complexity is lower than that of geometric active contours. Geometric active contours are represented implicitly in an Eulerian formulation as a level set of a higher dimensional function. The advantage of this method is that they can tackle topological changes without prior knowledge about the object. Moreover, snakes can be classified in terms of the type of image information used. If local feature such as edges is used, the method is called edge-based, if regional image information such as intensity, statistics color or texture is used; the method is called region-based.

The original parametric snakes are introduced by Kass et al. in [32] and the analysis of this method is given in section 6.1 in detail. Kass snake model is very sensitive to parameters and convergence depend on the initial curve. Furthermore, it has narrow capture range and fails to detect concave boundaries. To address some of the problems of the Kass snake model, Williams and Shah [57] proposed a new discrete approximation for the snake algorithms called the “greedy snake”. The evolution process of the greedy snake is different than that of Kass snake; It checks the neighborhood around each snake point and then moves to the point with the lowest energy. It also decreases the computational cost and provides a stable convergence. Additionally, to solve the problems of the snakes several methods have been developed. Gradient vector flow (GVF) snake [59], the balloon model [11], distance potential force [12] and dual active contour model [27] are some other examples of snake models. GVF snakes more efficient than other approaches. It is less sensitive to the initialization and it can move into boundary concavities. A level set formulation of active contours was proposed by Caselles et al. and Malladi et al. [6] to provide a solution and to address the limitations of parametric snakes. The evolution is independent of parameterization and based on the curve evolution theory. Later on Chan and Vese [9] introduced more effective active contour algorithm which is examined in Chapter 5.

6.1 Original Snake Model (Kass et al.)

A snake is a parametric curve which tries to move into a position where its energy is minimized. The following energy functional is introduced in [32].

$$E_{snk}(C) = \int_0^1 (E_{int}(C) + E_{ext}(C))\,\mathrm{d}s\ .$$

The internal energy of the snake is defined as

$$E_{int} = \frac{1}{2}\left(\alpha(s)|C_s|^2 + \beta(s)|C_{ss}|^2\right)\ , \tag{6.3}$$

where,

$C_s = \frac{\partial C}{\partial s}$: first order derivative of C that measures the elasticity of the snake.
$C_{ss} = \frac{\partial^2 C}{\partial s^2}$: second order derivative of C that measures the curvature of the snake.

The external energy of the snake is defined as $E_{ext} = E_{img} + E_{con}$. In order to simplify the analysis, the external constraint forces E_{con} can be disregarded.

The image energy E_{img} of the snake is defined as

$$E_{img} = -|\nabla I(x,y)|^2 \ , \tag{6.4}$$

or,

$$E_{img} = -|\nabla \left(G_\sigma(x,y) * I(x,y)\right)|^2 \ . \tag{6.5}$$

6.1.1 Euler-Lagrange Equations

The energy functional of the snake is

$$E_{snk}(C) = \int_0^1 \left(\frac{1}{2}\left(\alpha(s)|C_s|^2 + \beta(s)|C_{ss}|^2\right) + E_{ext}(C)\right) \mathrm{d}s \ . \tag{6.6}$$

To minimize this functional we can use the calculus of variations. The Euler-Lagrange equations are found by computing the Gateaux derivative of the energy functional. The Gateaux derivative (See Appendix) of $E_{snk}(C)$ in the direction Θ

$$DE_{snk}(C)[\Theta] = \lim_{\epsilon\to 0} \frac{E_{snk}(C+\epsilon\Theta) - E_{snk}(C)}{\epsilon} \ , \tag{6.7}$$

where $\Theta = (\eta(s), \zeta(s))$ and $C + \epsilon\Theta = (x + \epsilon\eta, y + \epsilon\zeta)$. Let $\alpha(s) = \alpha$ and $\beta(s) = \beta$ be constants. Then

$$\begin{aligned} E_{snk}(C+\epsilon\Theta) - E_{snk}(C) &= \int_0^1 \left(\frac{1}{2}\left(\alpha\left|\frac{\partial C}{\partial s} + \epsilon\frac{\partial \Theta}{\partial s}\right|^2 + \beta\left|\frac{\partial^2 C}{\partial s^2} + \epsilon\frac{\partial^2 \Theta}{\partial s^2}\right|^2\right) + E_{ext}(C+\epsilon\Theta)\right) \mathrm{d}s \\ &\quad - \int_0^1 \left(\frac{1}{2}\left(\alpha\left|\frac{\partial C}{\partial s}\right|^2 + \beta\left|\frac{\partial^2 C}{\partial s^2}\right|^2\right) + E_{ext}(C)\right) \mathrm{d}s \ . \end{aligned} \tag{6.8}$$

- $$\begin{aligned} \left|\frac{\partial C}{\partial s} + \epsilon\frac{\partial \Theta}{\partial s}\right|^2 &= \left(\frac{\mathrm{d}x}{\mathrm{d}s} + \epsilon\frac{\mathrm{d}\eta}{\mathrm{d}s}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}s} + \epsilon\frac{\mathrm{d}\zeta}{\mathrm{d}s}\right)^2 \\ &= \left(\frac{\mathrm{d}x}{\mathrm{d}s}\right)^2 + 2\epsilon\frac{\mathrm{d}x}{\mathrm{d}s}\frac{\mathrm{d}\eta}{\mathrm{d}s} + \epsilon^2\left(\frac{\mathrm{d}\eta}{\mathrm{d}s}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}s}\right)^2 + 2\epsilon\frac{\mathrm{d}y}{\mathrm{d}s}\frac{\mathrm{d}\zeta}{\mathrm{d}s} + \epsilon^2\left(\frac{\mathrm{d}\zeta}{\mathrm{d}s}\right)^2 . \end{aligned} \tag{6.9}$$

- $$\left|\frac{\partial^2 C}{\partial s^2}+\epsilon\frac{\partial^2\Theta}{\partial s^2}\right|^2 = \left(\frac{\mathrm{d}^2x}{\mathrm{d}s^2}+\epsilon\frac{\mathrm{d}^2\eta}{\mathrm{d}s^2}\right)^2+\left(\frac{\mathrm{d}^2y}{\mathrm{d}s^2}+\epsilon\frac{\mathrm{d}^2\zeta}{\mathrm{d}s^2}\right)^2$$
$$= \left(\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\right)^2+2\epsilon\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\frac{\mathrm{d}^2\eta}{\mathrm{d}s^2}+\epsilon^2\left(\frac{\mathrm{d}^2\eta}{\mathrm{d}s^2}\right)^2+\left(\frac{\mathrm{d}^2y}{\mathrm{d}s^2}\right)^2+2\epsilon\frac{\mathrm{d}^2y}{\mathrm{d}s^2}\frac{\mathrm{d}^2\zeta}{\mathrm{d}s^2}+\epsilon^2\left(\frac{\mathrm{d}^2\zeta}{\mathrm{d}s^2}\right)^2 . \tag{6.10}$$

The taylor expansion of the term $E_{ext}(C+\epsilon\Theta)$ is

$$\begin{aligned} E_{ext}(C+\epsilon\Theta) &= E_{ext}(x+\epsilon\eta, y+\epsilon\zeta) \\ &= E_{ext}+\epsilon\eta\frac{\partial E_{ext}}{\partial x}+\epsilon\zeta\frac{\partial E_{ext}}{\partial y}+O(\epsilon^2) . \end{aligned} \tag{6.11}$$

Substituting the above results (6.9), (6.10) and (6.11) in equation (6.8), we obtain

$$\begin{aligned} E_{snk}(C+\epsilon\Theta) \quad - \quad E_{snk}(C) = & \int_0^1 \frac{1}{2}\alpha\left(\left(\frac{\mathrm{d}x}{\mathrm{d}s}\right)^2+2\epsilon\frac{\mathrm{d}x}{\mathrm{d}s}\frac{\mathrm{d}\eta}{\mathrm{d}s}+\epsilon^2\left(\frac{\mathrm{d}\eta}{\mathrm{d}s}\right)^2+\left(\frac{\mathrm{d}y}{\mathrm{d}s}\right)^2+2\epsilon\frac{\mathrm{d}y}{\mathrm{d}s}\frac{\mathrm{d}\zeta}{\mathrm{d}s}+\epsilon^2\left(\frac{\mathrm{d}\zeta}{\mathrm{d}s}\right)^2\right)\mathrm{d}s \\ & + \int_0^1 \frac{1}{2}\beta\left(\left(\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\right)^2+2\epsilon\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\frac{\mathrm{d}^2\eta}{\mathrm{d}s^2}+\epsilon^2\left(\frac{\mathrm{d}^2\eta}{\mathrm{d}s^2}\right)^2+\left(\frac{\mathrm{d}^2y}{\mathrm{d}s^2}\right)^2+2\epsilon\frac{\mathrm{d}^2y}{\mathrm{d}s^2}\frac{\mathrm{d}^2\zeta}{\mathrm{d}s^2}+\epsilon^2\left(\frac{\mathrm{d}^2\zeta}{\mathrm{d}s^2}\right)^2\right)\mathrm{d}s \\ & + \int_0^1 \left(E_{ext}+\epsilon\eta\frac{\partial E_{ext}}{\partial x}+\epsilon\zeta\frac{\partial E_{ext}}{\partial y}\right)\mathrm{d}s \\ & - \int_0^1 \left(\frac{1}{2}\left(\alpha\left(\left(\frac{\mathrm{d}x}{\mathrm{d}s}\right)^2+\left(\frac{\mathrm{d}y}{\mathrm{d}s}\right)^2\right)+\beta\left(\left(\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\right)^2+\left(\frac{\mathrm{d}^2y}{\mathrm{d}s^2}\right)^2\right)\right)+E_{ext}\right)\mathrm{d}s , \end{aligned} \tag{6.12}$$

$$\begin{aligned} \lim_{\epsilon\to 0}\frac{E_{snk}(C+\epsilon\Theta)-E_{snk}(C)}{\epsilon} &= \int_0^1 \left(\alpha\left(\frac{\mathrm{d}x}{\mathrm{d}s}\frac{\mathrm{d}\eta}{\mathrm{d}s}+\frac{\mathrm{d}y}{\mathrm{d}s}\frac{\mathrm{d}\zeta}{\mathrm{d}s}\right)+\beta\left(\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\frac{\mathrm{d}^2\eta}{\mathrm{d}s^2}+\frac{\mathrm{d}^2y}{\mathrm{d}s^2}\frac{\mathrm{d}^2\zeta}{\mathrm{d}s^2}\right)\right)\mathrm{d}s \\ &+ \int_0^1 \left(\eta\frac{\partial E_{ext}}{\partial x}+\zeta\frac{\partial E_{ext}}{\partial y}\right)\mathrm{d}s . \end{aligned} \tag{6.13}$$

Since $DE_{snk}(C)[\Theta] = 0$ for all admissible Θ, (6.13) yields the following pair of equations

$$\int_0^1 \left(\alpha\frac{\mathrm{d}x}{\mathrm{d}s}\frac{\mathrm{d}\eta}{\mathrm{d}s}+\beta\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\frac{\mathrm{d}^2\eta}{\mathrm{d}s^2}+\eta\frac{\partial E_{ext}}{\partial x}\right)\mathrm{d}s = 0 , \tag{6.14}$$

$$\int_0^1 \left(\alpha\frac{\mathrm{d}y}{\mathrm{d}s}\frac{\mathrm{d}\zeta}{\mathrm{d}s} + \beta\frac{\mathrm{d}^2y}{\mathrm{d}s^2}\frac{\mathrm{d}^2\zeta}{\mathrm{d}s^2} + \zeta\frac{\partial E_{ext}}{\partial y}\right)\mathrm{d}s = 0\ . \tag{6.15}$$

Using the integration by parts rule (see Appendix) in equation (6.14), we obtain

- $$\int_0^1 \alpha\frac{\mathrm{d}x}{\mathrm{d}s}\frac{\mathrm{d}\eta}{\mathrm{d}s} = \left[\alpha\frac{\mathrm{d}x}{\mathrm{d}s}\eta\right]_0^1 - \int_0^1 \eta\frac{\mathrm{d}}{\mathrm{d}s}\left(\alpha\frac{\mathrm{d}x}{\mathrm{d}s}\right)\mathrm{d}s\ . \tag{6.16}$$

- $$\int_0^1 \beta\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\frac{\mathrm{d}^2\eta}{\mathrm{d}s^2}\mathrm{d}s = \left[\beta\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\frac{d\eta}{\mathrm{d}s}\right]_0^1 - \left[\frac{d}{\mathrm{d}s}\left(\beta\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\right)\eta\right]_0^1 + \int_0^1 \eta\frac{d^2}{\mathrm{d}s^2}\left(\beta\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\right)\mathrm{d}s\ . \tag{6.17}$$

Since we are interested with a closed curve

$$\eta(1) - \eta(0) = 0 \text{ and } \zeta(1) - \zeta(0) = 0\ \ .$$

So equation (6.14) can be written as

$$\int_0^1 \left(-\frac{\mathrm{d}}{\mathrm{d}s}\left(\alpha\frac{\mathrm{d}x}{\mathrm{d}s}\right) + \frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\beta\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\right) + \frac{\partial E_{ext}}{\partial x}\right)\eta\mathrm{d}s = 0\ . \tag{6.18}$$

From the fundamental lemma of the calculus of variations (see Appendix), we obtain the following Euler-Lagrange equation

$$-\frac{\mathrm{d}}{\mathrm{d}s}\left(\alpha\frac{\mathrm{d}x}{\mathrm{d}s}\right) + \frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\beta\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\right) + \frac{\partial E_{ext}}{\partial x} = 0\ . \tag{6.19}$$

Similarly, equation (6.15) yields

$$-\frac{\mathrm{d}}{\mathrm{d}s}\left(\alpha\frac{\mathrm{d}y}{\mathrm{d}s}\right) + \frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\beta\frac{\mathrm{d}^2y}{\mathrm{d}s^2}\right) + \frac{\partial E_{ext}}{\partial y} = 0\ . \tag{6.20}$$

6.1.2 Numerical Approximations

In order to solve the Euler-Lagrange equations (6.19)and (6.20) numerically, an artificial time paremeter *t* is introduced and the distributed steepest descent formulation is applied

$$\gamma \frac{\partial C}{\partial t} = \frac{\partial}{\partial s}\left(\alpha \frac{\partial C}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta \frac{\partial^2 C}{\partial s^2}\right) - \frac{\partial E_{ext}}{\partial C} \ , \tag{6.21}$$

or simply as

$$\gamma C_t = \alpha C_{ss} - \beta C_{ssss} - \nabla E_{ext}(C) \ , \tag{6.22}$$

$$\gamma \frac{\partial C}{\partial t} = F_{int}(C) + F_{ext}(C) \ , \tag{6.23}$$

where $F_{int}(C) = \alpha C_{ss} - \beta C_{ssss}$ and $F_{ext}(C) = -\nabla E_{ext}(C)$. We can discretize the model by using finite differences

$$C_i^n = (x_i^n, y_i^n) = (x(ih, n\Delta t), y(ih, n\Delta t)) \ ,$$

$$\frac{\partial C}{\partial t} = \frac{C_i^n - C_i^{n-1}}{\Delta t} \ ,$$

$$\frac{\partial C}{\partial s} = \frac{C_i^n - C_{i-1}^n}{h} \ ,$$

$$\frac{\partial^2 C}{\partial s^2} = \frac{C_{i-1}^n - 2C_i^n + C_{i+1}^n}{h^2} \ .$$

A compact matrix form is obtained as

$$\frac{C^n - C^{n-1}}{\tau} = AC^n + F_{ext}(C^{n-1}) \ , \tag{6.24}$$

where $A : mxm$ petadiagonal banded matrix (m is the number of sample points) and $\tau = \frac{\Delta t}{\gamma}$. The above equation (6.24) can be solved iteratively by matrix inversion

$$C^n = (I - \tau A)^{-1}(C^{n-1} + \tau F_{ext}(C^{n-1})) \ . \tag{6.25}$$

The inverse of the matrix $(I - \tau A)$ can be calculated efficiently by LU decomposition. The new snake points are defined as

$$x^n = (I - \tau A)^{-1} \left(x^{n-1} - \tau \frac{\partial E_{img}}{\partial x^{n-1}} \right) , \tag{6.26}$$

$$y^n = (I - \tau A)^{-1} \left(y^{n-1} - \tau \frac{\partial E_{img}}{\partial y^{n-1}} \right) . \tag{6.27}$$

6.1.3 Experimental Results

In this section, the Kass snake model, examined in detailed in Section 6.1.1 and Section 6.1.2 have been implemented. In the experiments, we take $\alpha = 0.05$ and $\beta = 0.0005$. In Figure 6.1, an example of snake evolution is illustrated. The initial snake is located around the object and shown in Figure 6.1 (a); the blue dots marks initial location of the snake points and the yellow curve, interpolation of the points gives the initial curve. By minimizing the snake energy, it is converged the boundary of the object and the final location of the snake after 618 iterations is shown in Figure 6.1 (b). In Figure 6.2, the segmentation result of snake evolution is demonstrated. Figure 6.2 (a) shows initial position of the snakes and the object to be detected. The snake reached the boundary of the desired object after 399 iterations and the final result illustrated in Figure 6.2 (b). One of the limitations of the active contour is shown in Figure 6.3. The initial snake is located around the object as a circle, illustrated in Figure 6.3 (a). The result of evolution is not good enough in Figure 6.3 (b) and Figure 6.3 (c); the snake cannot move concavity in boundary. After many tests for the values of α and $\beta = 0.05$, a successful boundary localization is obtained; but it consumes much time. In Figure 6.4 the snake evolution results of a man image are given. Figure 6.4 (a) the initial curve located in the center of the image and the result after 3000 iterations seen in Figure 6.4 (b) shows that the snake is not able to find the boundaries. When we manually place the initial curve near the boundary, we obtain better result but it still does not find the concave boundaries. The initial curves and the results after 2000 and 5000 iterations are given in Figure 6.4 (c)-(d) and Figure 6.4 (e)-(f), respectively. Figure 6.5 illustrates the comparison of the classic active contours and the Chan-Vese active contour model. The results obtained using the classic snakes based on edge function are shown in Figure 6.5 (a)-(b), while the results obtained using the Chan-Vese model are presented in Figure 6.5 (c)-(d). The classic snakes are not be successful to find the smooth contours for different intensities; on the other hand the Chan-Vese algorithm automatically detects the smooth contours.

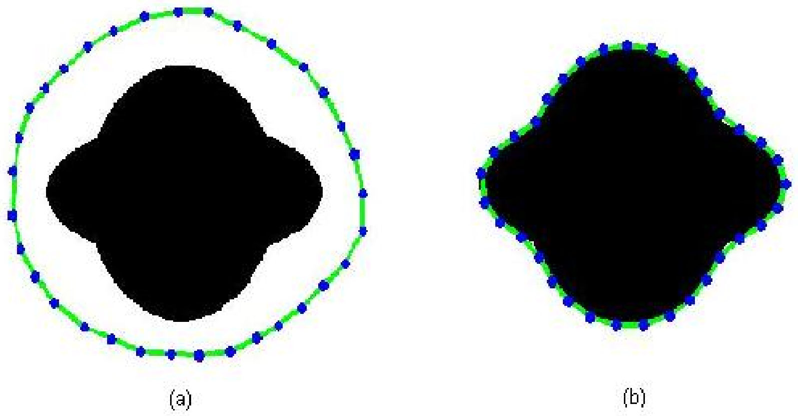

Figure 6.1: Contour detection of an image via the active contour model. (a) The iniitial situation of the snake and the object. (b) The final situation of the snake using the Kass snake model after 618 iterations.

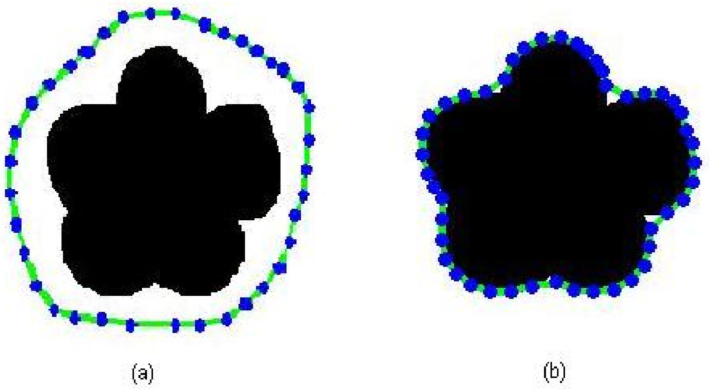

Figure 6.2: The snake evolution result. (a) The initial snake and the object to be detected. (b) The final snake after 399 iterations.

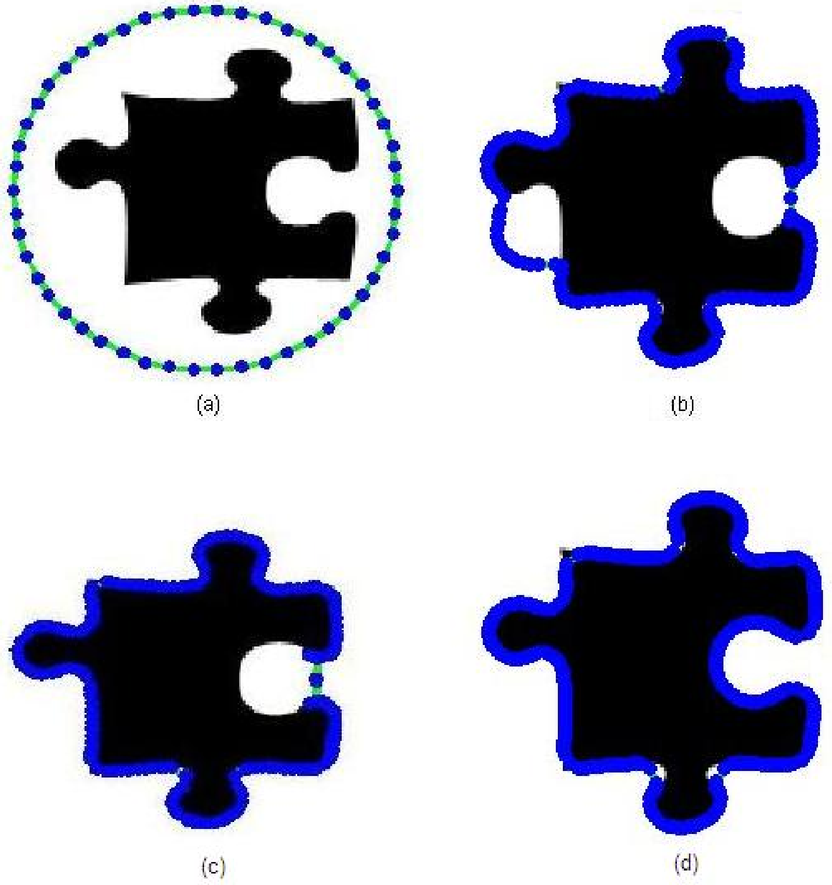

Figure 6.3: The snake evolution for boundary concavity. (a) The initial snake. (b) The result of Kass snake model; the snake cannot detect the boundary well after 1500 iterations. (c) The result after 2000 iterations; the snake is still not able to move into concavity after 2000 iterations. (d) Finally, the snake reaches the boundary concavity with $\alpha = 0.1$, $\beta = 0.05$ after 756 iterations.

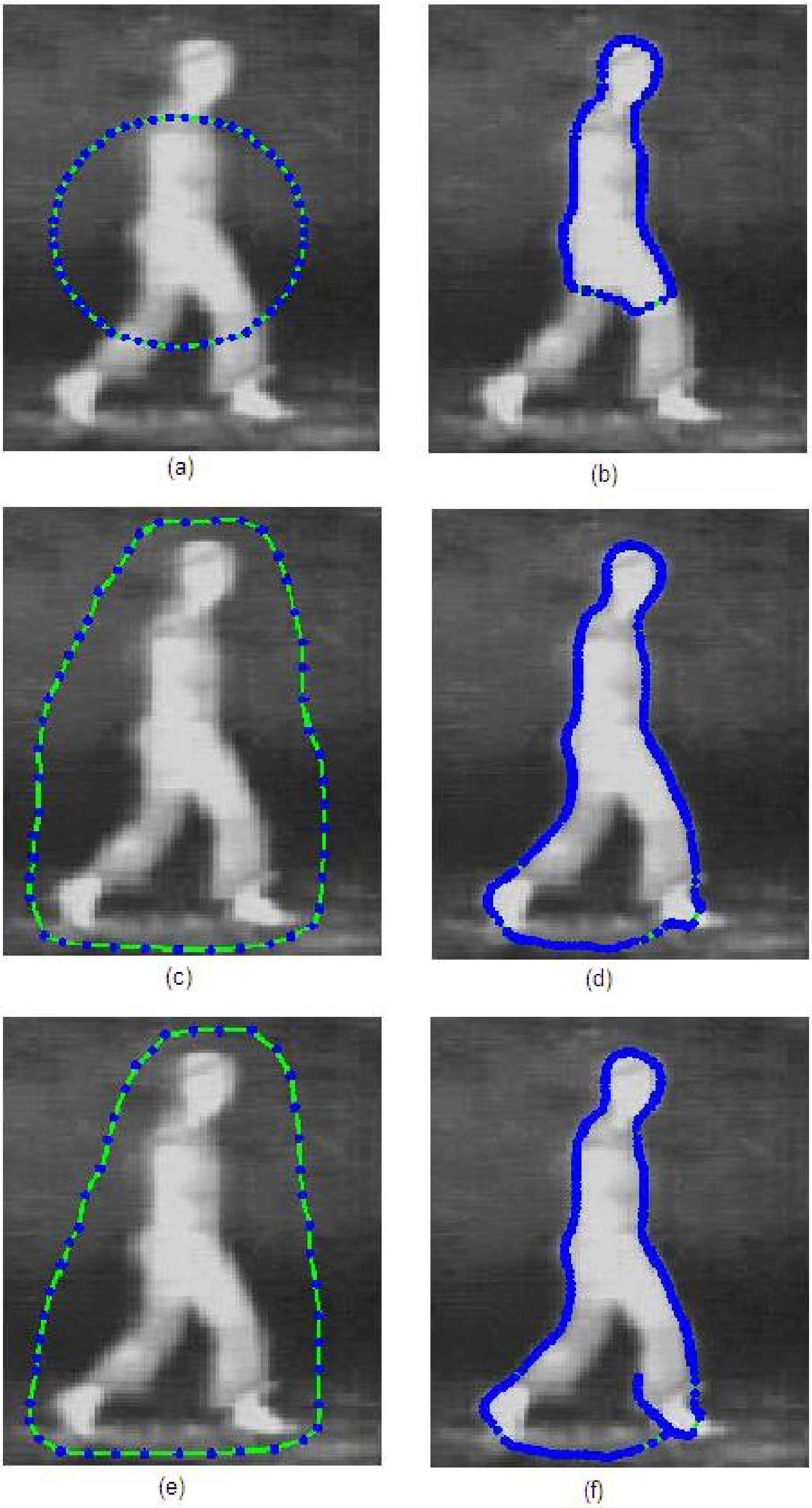

Figure 6.4: The snake evolution results.

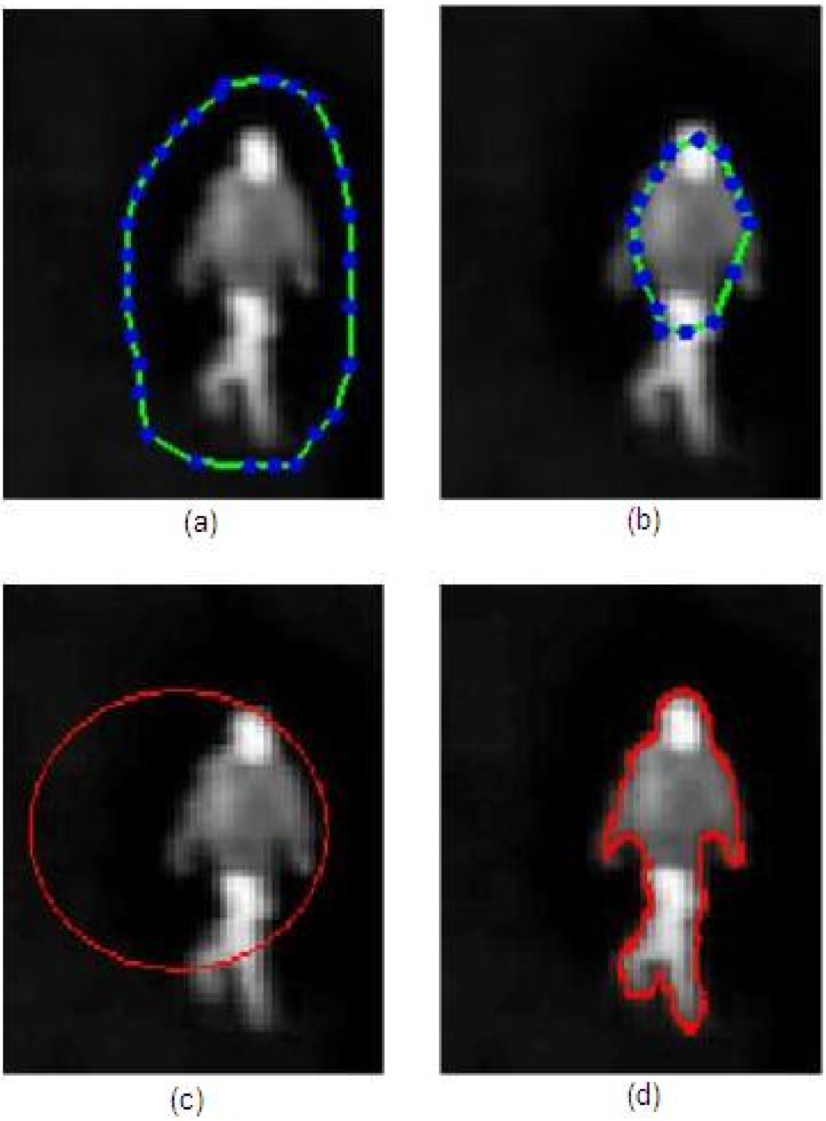

Figure 6.5: Comparison the classic snakes with the Chan-Vese active contour model.

CHAPTER 7

COMPARISON OF THE MODELS

It is not easy to make a quantitative comparison among the methods explained in previous chapters. All these methods have different applications and each of them is successful in special fields. So, in general, we cannot say that one method is better than the other. On the other hand, one can compare the methods according to their similarities and differences and point out their strong and weak characteristics for different image types. Before making a comparison, we have classified the images and determined image types. First, we have grouped images according to their contained objects. All methods are applied on a basic image which contains only one object and the results of each algorithm are given in Figure 7.1. Two methods are applied together, e.g.,the Perona-Malik and the Chan-Vese or the Ambrosio-Tortorelli and the Chan-Vese, on the same image and the results are given in Figure 7.1. As seen from the results, when the Chan-Vese and the snake algorithm are applied with the Ambrosio-Tortorelli algorithm, we get better segmentation results. Next, we have classified images with respect to their texture properties and compared the methods. The Perona-Malik, the Ambrosio-Tortorelli and the Chan-Vese algorithms are applied on a basic textured image given in Figure 7.3. In this comparison, we do not deal with the snake algorithms, because it is not appropriate for this type of images. It has been observed that the Chan-Vese algorithm was not able to find the boundaries of the textured image. The results of the Chan-Vese model after the Perona-Malik and the Ambrosio-Tortorelli algorithms are illustrated in Figure 7.4. However, when the Perona-Malik or the Ambrosio-Tortorelli method is applied before the Chan-Vese model, the boundaries are detected successfully. As a result, we can conclude that if a nonlinear diffusion method is applied before any segmentation algorithm, more successful results can be obtained.

Table 7.1: Comparison of the models with respect to number of the objects in an image.

Model	**Image Type**	**Noise reduction**	**Time**
PM	Mid-complex, low-complex and basic images	High	140.3299 (150 iterations)
AT	Mid-complex, low-complex and basic images	High	31.5321 (150 iterations)
CV	Low-complex and basic images	Not applicable	26.8008 (150 iterations)
Snakes	Basic images	Not applicable	2.8715 (2000 iterations)
PM + CV	Low-complex images	High	169.4669 (300 iterations)
PM + Snake	Basic images	High	144.0586 (2150 iterations)
AT + CV	Low-complex and basic images	High	42.4775 (300 iterations)
AT + Snake	Basic images	High	167.3603 (2150 iterations)

Model:

PM: Perona-Malik model

AT: Ambrosio-Tortorelli model

CV: Chan-Vese model

Snake: Active Contour model

Image Type: Images can be classified according to their phase.

- Basic images contain one simple object.
- Low-complex images contain two or more simple objects.
- Mid-complex images contain several objects.
- High-complex images contain large number of small objects.

For high-complex images, the segmentation can be meaningless.

Noise Reduction: For the noise we keep in view the previous results given in the chapters.

Time: All algorithms are tested on the same basic image and the results given in Figure 7.1. Figure 7.2 shows the results of AT, CV and Snake algorithm with PM algorithm together. The computational time of each algorithm is calculated and given in Table 7.1. In the snake algorithm the time is calculated after 2000 iterations, however it consumes the minimal time.

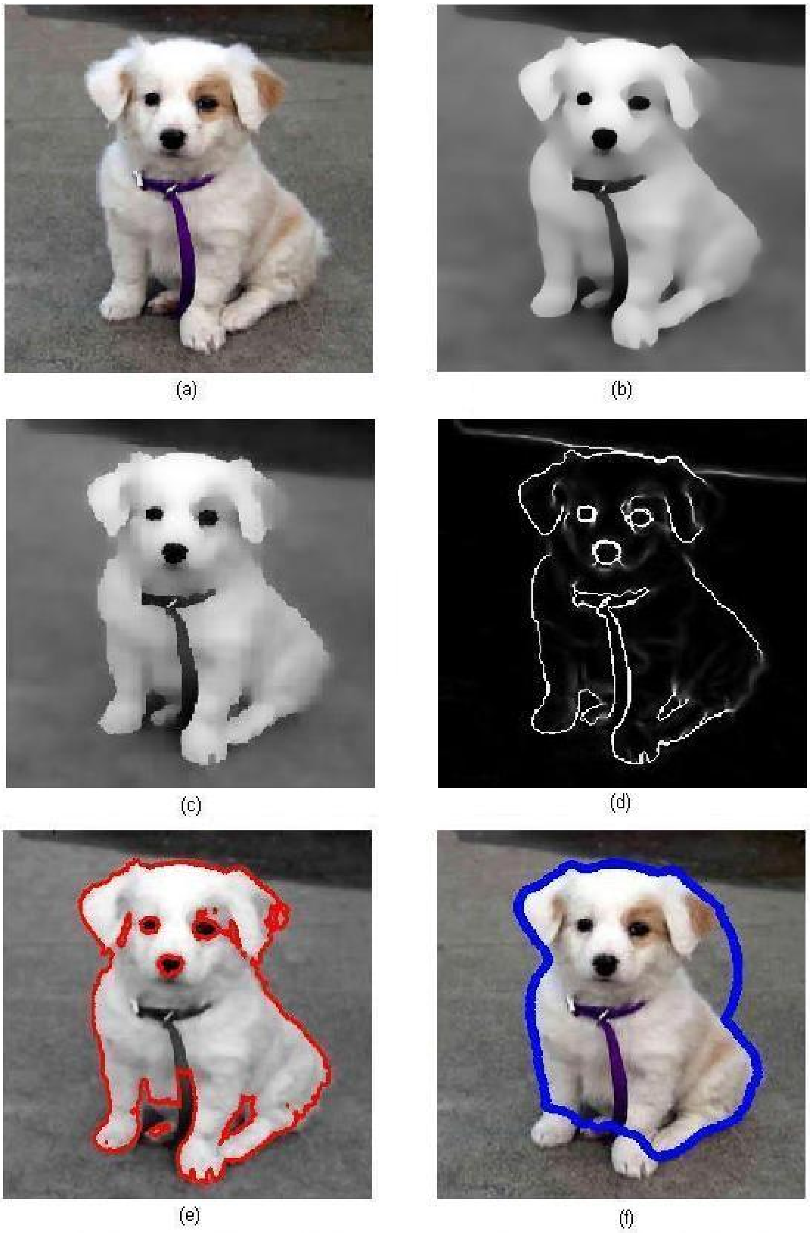

Figure 7.1: Comparison of the models. (a) The original image. (b) The result of PM model after 150 iteratios. (c)-(d) The results of the AT model after 150 iterations for $\alpha = 2, \beta = 0.1$, $\rho = 0.002$. (e) The result of CV model after 150 iterations. (f) The resuslt of snake model after 2000 iterations for $\alpha = 0.12, \beta = 0.04$.

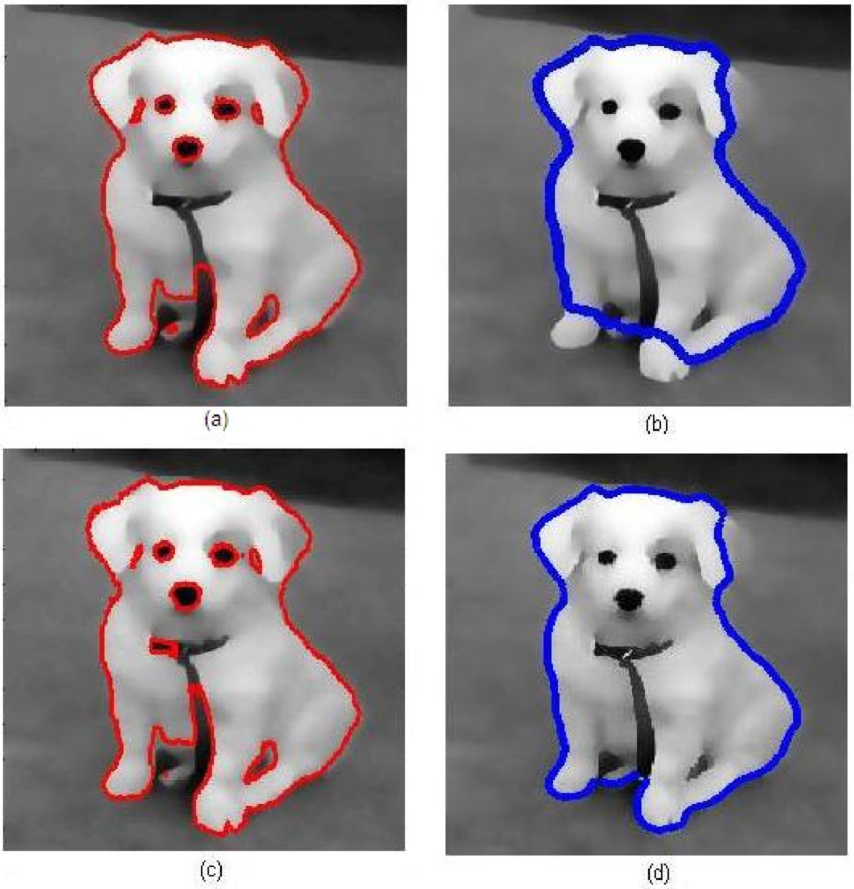

Figure 7.2: (a)-(b) The results of CV and snake models after PM model, respectively. (c)-(d)The results of CV and snake models after AT model, respectively.

Table 7.2: Comparison of the models with respect to texture.

Model	Image Type	Noise reduction	Time
PM	Mid-textured and low-textured images	High	123.2241 (100 iterations)
AT	High-textured, mid-textured and low-textured images	High	19.9300 (100 iterations)
CV	Failed	Not applicable	23.3991 (100 iterations)
Snakes	-	-	-
PM + CV	Low-textured images	High	146.2982 (200 iterations)
AT + CV	Low-textured images	High	33.0520 (200 iterations)

Image Type: We classify the images according to textured as

- Low-textured images
- Mid-textured images
- High-textured images

Noise Reduction: For the noise we keep in view the previous results given in the chapters.

Time: All algorithms are tested on the same low-textured image and the results given in Figure 7.3. Figure 7.4 shows the results of AT and CV algorithm with PM algorithm together. The computational time of each algorithm is calculated for a low-textured image and given in Table 7.2.

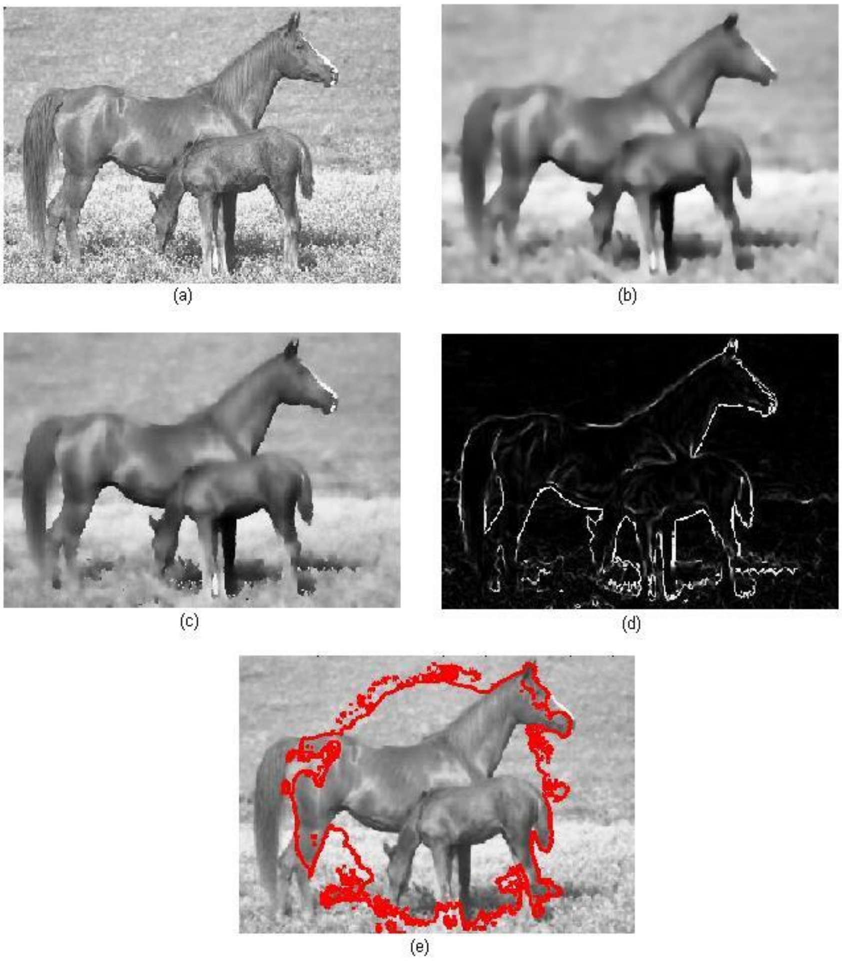

Figure 7.3: (a) The original image. (b) The result of PM model after 100 iteratios. (c)-(d) The results of the AT model after 100 iterations for $\alpha = 2, \beta = 0.2, \rho = 0.001$. (e) The result of CV model after 100 iterations.

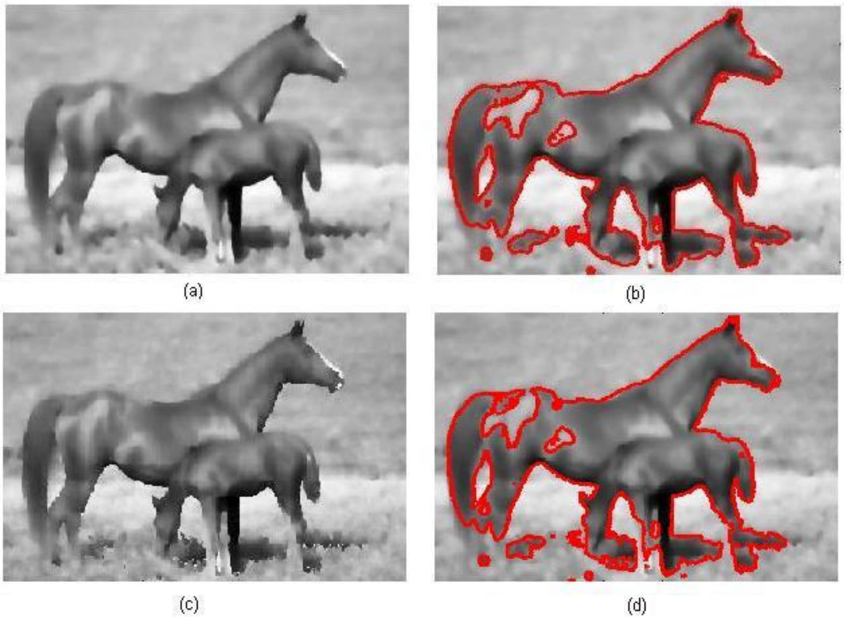
(a)
(b)
(c)
(d)

Figure 7.4: (a) The result of PM model. (b) The result of CV model with PM model. (c) The result of AT model. (d) The result of CV model with AT model.

CHAPTER 8

CONCLUSION

Image segmentation and smoothing applications are the most widely studied issues in image processing community. In the literature, there are several approaches proposed to deal with segmentation and smoothing problems. In this study, the image segmentation and smoothing techniques based on PDEs, have been examined. At first, linear diffusion equation based methods have been used for reducing noise, while preserving shapes in an image. To overcome the difficulties of the linear diffusion methods, Perona and Malik proposed a nonlinear diffusion method, also known as anisotropic diffusion. It is a well-known denoising methodology and one of the early developed PDE-based methods. The numerical analysis of the Perona-Malik model and experimental results have been presented in this work. The Mumford-Shah functional which is the most widely studied model in image processing deals with both segmentation and denoising problems. The key idea is to minimize an energy functional in order to obtain a homogeneous representation of the image and detect the boundaries of the smooth regions. Unfortunately, the Mumford-Shah functional is complicated and nonconvex, so it has numerous local minima. To handle the problems of the Mumford-Shah functional, several approaches have been proposed. The details and some approximations of this model have been described within the scope of this thesis. Two of such approximations of the Mumford-Shah functional are the Ambrosio-Tortorelli model and the Chan-Vese active contours model. They are both used for segmenting an image, but which one is used depends on the kind of the image under investigation. If we are interested in an image in which specific objects are significant and we want to find the boundaries of objects, we may prefer to use the Chan-Vase model. If we deal with an image in which all contours are important and we would like to find an approximation of the initial image composed of homogenous regions, the Ambrosio-Tortorelli approximation of the Mumford-Shah functional can be used. In this thesis, the Ambrosio-Tortorelli model has been described extensively. Derivation of the Euler-Lagrange equations and numerical implementations of the model have been shown in detail. The Chan-Vese active contours model based on the Mumford-Shah functional and level set method has been presented, as well. Mathematical analysis and numerical approximation of the energy functional have been given. Moreover, another commonly used edge-based segmentation technique, "Snakes: Active Contour Models" is studied. An active contour is defined as a curve or a surface within an image and it detects the boundary of a desired object. Classical Snakes introduced by Kass et al. is described and numerical approximations of this model are presented. During this study, MATLAB codes have been developed to implement all image segmentation methods investigated within this thesis. The developed programs of these techniques have been applied on several synthetic and real images and the results have been interpreted. Comparing the results, we can conclude that the performance of the Chan-Vese model is better than the classic active contours. The Chan-Vese active contour method has successfully found the boundaries of more than one object and it is more robust to noise. The evolution of the Chan-Vese model does not depend on location of initial curve. It can be anywhere on the image domain, but in the snakes the curve must be initialized near the object focused on.

To make a comparison, all methods are applied on the same type of images. Evaluating the performance of the algorithms, two methods have been applied together on the same images. The experimental results have shown that the boundary localization performed after the Ambrosio-Tortorelli model implementation is better than that of the Perona-Malik model. Comparing the methods examined in this study, the fastest one is the classical snake method; but it has some limitations. To deal with the limitations of the classical snakes some regularization should be made. Statistical information extracted from images with shape information can be used to learn the possible shape of the object, before the snake method is applied. Therefore, the shape of the initial curve had better be chosen similar to the object to be detected and controlled during the process. This may be useful for medical image segmentation applications.

Considering limitations and results of this study, some of the issues identified for future work are as follows.

- In segmentation and edge detection, to verify the results a criteria should be determined.

In this thesis, any quantitative analysis is not applied to the segmentation results. Verification of the segmentation results is considered, qualitatively. A methodology can be developed or any existing method can be used to quantify the results. A ground truth of the desired image can be obtained by human or somehow and it can be compared with segmentation result.

- The importance of the scaling parameters (e.g., α, β) is understood from the experimental results given in previous chapters, especially for the Ambroiso-Tortorelli and the classical snake methods. Many trials have been made to find the best values of them. If we want to segment a new image, we should run the segmentation algorithms many times to find the appropriate α and β values. However, testing many values in an order does not ensure the best result. By examining the effect of the parameters to the model and the relation between the parameters and the image, some formulas for α and β may be proposed.
- As seen in some experiments of the Ambrosio- Tortorelli model, the strong noise cannot be eliminated and after many iterations some details get blurred. A new regularization may be developed to remove noise completely, while keeping edges.
- Generally, all these methods are applied to the basic images because it is hard to apply the segmentation methods to the complex images which contain large numbers of small objects. Also the size of the image is significant for the computation time. The sizes of the images used in the experiments are usually small. But for the images with big region sizes the computations may take too much time. If possible, speeds of the algorithms may be increased.

REFERENCES

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

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

[3] Ambrosio, L. and Tortorelli, V., “On the approximation of functionals depending on jumps by elliptic functionals via Γ-convergence,” *Commun. Pure Appl. Math.,* vol. 43, no. 8, pp. 999-1036, 1990.

[4] Aubert, G. and Kornprobst, P., *Mathematical Problems in Image Processing.* Springer-Verlag, New York, 2002.

[5] Biemond, J., Lagendijk, R.L., and Mersereau, R.M., “Iterative methods for image deblurring,’ *Proceedings. of the IEEE,* vol.78, no. 5, pp. 856-883, May 1990.

[6] Caselles, V., Catte, F., Coll, T. and Dibos, F., “A geometric model for active contours,” *Numerische Mathmatik,* vol. 66, no. 1, pp. 1-31, 1993.

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

[8] Chambolle, A., De Vore, R.D., Lee, N., and Lucier, B.J. “Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage,” *IEEE Transactions on Image Processing,* vol. 7, no. 3, pp. 319-335, 1998.

[9] Chan, T. and Vese, L., “Active contours without edges,” *IEEE Trans. on Image Processing,* vol. 10, no. 2, pp. 266-277, 2001.

[10] Chan, T. and Vese, L., “Active Contour and Segmentation Models Using Geometric PDE’s for Medical Imaging, UCLA Computational and Applied Mathematics Reports, December 2000.

[11] Cohen, L.D., “On active contour models and balloons,” *CVGIP: Image Understanding,* vol. 53, no. 2, pp. 211-218, 1991.

[12] Cohen, L.D. and Cohen, I., “Finite-element methods for active contour models and balloons for 2-D and 3-D images,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 15, no. 11, pp. 1131-1147, 1993.

[13] Dambreville, S., “Statistical and Geometric Methods for Shape-Driven Segmentation and Tracking,” Ph.D. dissertation, School of Electrical and Computer Engineering, Georgia Institute of Technology, April 2008.

[14] Dascal, L.,“Well-posedness and maximum principle for PDE based models in image processing,” Ph.D. dissertation, Tel Aviv University, March 2006.

[15] Demant, C., Streicher-Abel, B., Waszkewitz, P., *Industrial Image Processing: Visual Quality Control in Manufacturing.* Springer, Berlin, Germany, 1999.

[16] De Vore, R.A., Jawerth, B. and Lucier, B.J., “Image compression through wavelet transform coding,” *IEEE Transactions on Information Theory,* vol. 38, no. 2, pp. 719-746, March 1992.

[17] Donoho, D.L. and Johnstone, M., “Adapting to unknown smoothness via wavelet shrinkage,” *Journal of the American Statistical Association,* vol. 90, no. 432, pp. 1200-1224, 1995.

[18] Erdem, E., Erdem, A. and Tari S, “Edge strength functions as shape priors in image segmentation,” *EMM-CVPR,* pp. 490-502, 2005.

[19] Erdem, E., Tari, S., “Mumford-Shah regularizer with contextual feedback,” *JMIV,* 2007.

[20] Erdem, E., Yilmaz, A. Sancar and Tari, S., “Mumford-Shah regularizer with spatial coherence,” *SSVM,* pp. 545- 555, 2007.

[21] Fabijanska, A., Strzecha, K. and Sankowski, D., “Determination of image segmentation quality,” *IEEE 9th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, Proceedings of the 9th International Conference CADSM’2007,* pp. 439-441, 2007.

[22] Gao, S., Bui, T.D., “Image segmentation and selective smoothing by using Mumford-Shah model,” *IEEE Trans. Image Process.,* vol. 14, no. 10, pp. 1537-1549, Oct. 2005.

[23] Geman, S. and Geman, D., “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 6, pp. 721-741, 1984.

[24] Gimel’farb, G.L., *Image Texture and Gibbs Random Fields.* Kluwer Academic Publishers, Dordrecht, 1999.

[25] Gonzales, R.C., Woods, R.E., *Digital Image Processing.* Prentice Hall, 2007.

[26] Gudivada, V.N., Raghavan, V.V., “Content-based image retrieval systems,” *IEEE Computer,* vol. 28, no.9, pp. 18-22, 1995.

[27] Gunn, S.R. and Nixon, M.S., “A robust snake implementation: A dual active contour,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 19, no. 1, pp. 63-68, 1997.

[28] Hollister, S. J., Levy R. A., Chu, T. M., Halloran, J. W. and Feinberg, S. E., “An image-based approach for designing and manufacturing craniofacial scaffolds,” *International Journal of Oral and Maxillofacial Surgery,* vol. 29, pp. 67-71, 2000.

[29] Jung, C.R., and Scharcanski, J., “Robust watershed segmentation using wavelets,” *Image and Vision Computing,* vol. 23, no. 7, pp. 661-669, 2005.

[30] Javidi, B. (ed.), *Image Recognition and Classification: Algorithms, Systems, and Applications.* CRC Press, Marcel-Dekker, New York, 2002.

[31] Ji, L. and Yan, H., “Attractable snakes based on the greedy algorithm for contour extraction,” *Pattern Recognit.,* vol. 33, no. 4, pp. 791-806, 2002.

[32] Kass M., A. W. and Terzopoulos, D., “Snakes: active contour models,” *Int. Journal of Computer Vision,* vol. 1, pp. 321-331, 1987.

[33] Kovacevic, R. and Zhang, Y. M., “Real-time image processing for monitoring of free weld pool surface,” *the ASME Journal of Manufacturing Science and Engineering,* vol.119, no. 1, pp. 161-169, May 1997.

[34] Li, S. Z., *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, 1995.

[35] Malgouyres, F., “Combining total variation and wavelet packet approaches for image deblurring,” *IEEE Workshop on Variational and Level Set Methods,* IEEE Press, Piscataway, NJ, pp. 57-64, 2001.

[36] Malladi, R., Sethian, J.A., and Vemuri, B.C., “Shape modeling with front propagation: Level set approach,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 17, no. 2, pp. 158-175, 1995.

[37] Malladi, R., *Geometric Methods in Bio-medical Image Processing.* Springer, 2002.

[38] Martin, D., Fowlkes, C., Tal, D., and Malik, J., “A dataset of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” technical report, Computer Science Division (EECS), University of California, Berkeley, 2001.

[39] Mikula, K., “Image processing with partial differential equations,” *Modern Methods in Scientific Computing and Applications (A.Bourlioux and M.Gander, Eds.), NATO Science Ser.II,* vol. 75, pp. 283-322, 2002.

[40] Morel, J.-M. and Solimini, S., *Variational Methods for Image Segmentation.* Birkhauser, 1994.

[41] Mumford, D. and Shah, J., “Optimal approximation by piecewise smooth functions and associated variational problems,” *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577-684, 1989.

[42] Osher, S., Sethian, J.A., “Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *Journal Comp. Physics,* vol. 79, pp. 12-49, 1988.

[43] Paragios, N. and Deriche, R., “Geodesic active contours and level sets for the detection and tracking of moving objects,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 22, no. 3, pp. 266-280, 2000.

[44] Perona, P. and Malik, J., “Scale space and edge detection using anisotropic diffusion,” *IEEE Trans. Patt. Anal. Mach. Intell.,* vol. 12, pp. 629-639, 1990.

[45] Shah, J., “Segmentation by nonlinear diffusion,” *IEEE Conference on Computer Vision and Patern Recognition,* pp. 202-207, 1991.

[46] Shah, J., “A common framework for curve evolution, segmentation and anisotropic diffusion,” *IEEE Conference on Computer Vision and Patern Recognition,* pp. 136-142, June 1996.

[47] Shen, J., "Inpainting and the fundamental problem of image processing," *SIAM News,* vol. 36, no. 5, June 2003.

[48] Tari, S., Shah, J. and Pien. H. "Extraction of shape skeletons from grayscale images," *CVIU,* vol. 66, no. 2, pp. 133-146, May 1997.

[49] Tiilikainen, N. P., "A comparative study of active contour snakes," technical report, Department of Computer Science, University of Copenhagen (DIKU), 2007.

[50] Tsai, A., Yezzi, A., and Willsky, A., "Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Trans. Image Process.,* vol. 10, no. 8, pp. 1169- 1186, Aug. 2001.

[51] Umbaugh, Scott E., "Computer Imaging: Digital Image Analysis and Processing," *The CRC Press, Boca Raton, FL,* January 2005. ISBN: 0-84-932919-1.

[52] Vese, L., Chan, T., "A multiphase level set framework for image segmentation using the Mumford and Shah model," *Inter. Journal Comp. Vision,* vol. 50, no. 3, pp. 271-293, July 2002.

[53] Voci, F., Eiho, S., Sugimoto N. and Sekiguchi,H., "Estimating the Gradient Threshold in the Perona-Malik Equation," *IEEE Signal Processing Magazine,* pp. 39-46, May 2004.

[54] Wang, L., Hu, W. and Tan, T., "Recent developments in human motion analysis," *Pattern Recognition,* vol. 36, no. 3, pp. 585-601, March 2003.

[55] Weickert, J. "Theoretical foundations of anisotropic in image processing," *Comput. Suppl.,* vol. 11, pp. 221-236, 1996.

[56] Weickert, J. *Anisotrophic Diffusion in Image Processing.* Stutgart, Germany: Taeubner, 1998.

[57] Williams, D.J. and Shah M., "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding,* vol. 55, no. 1, pp. 14-26, 1992.

[58] Wysk, R.A., Yang, N.S., Joshi, S., "Detection of deadlocks in flexible manufacturing cells," *IEEE Transactions on Robotics and Automation,* vol. 7, no. 6, pp. 853-859, December 1991.

[59] Xu, C., Prince, J.L., "Snakes, shapes, and gradient vector flow," *IEEE Trans. on Image Process,* vol. 7, no. 3, pp. 359-369, 1998.

[60] Xu, C., Prince, J.L., "Gradient vector flow: A new external force for snakes," *IEEE Proc. Conf. on Comput. Vis. Patt. Recog. (CVPR)*, pp. 66-71, 1997.

[61] Xu, C., Pham, D. L. and Prince, J. L., "Image Segmentation Using Deformable Models," *Handbook of Medical Imaging: Vol. 2. Medical Image Processing and Analysis,* SPIE Press, 2000.

[62] Yezzi, A., Jr. Kichenassamy, S., Kumar, A., Olver, P. and Tannenbaum, A., "A geometric snake model for segmentation of medical imagery," Technical Report, Department of Electrical Engineering, University, Minneapolis, MN, July 1995.

[63] Zhou, H. M., "Wavelet Transforms and PDE Techniques in Image Compression," Ph.D. dissertation, University of California, Los Angeles, June 2000.

APPENDIX A

Some Mathematical Results

Definition A.1 ***(Gateaux Derivative)*** *Suppose X and Y are locally convex topological vector spaces (for example Banach spaces), $U \subset X$ is open and $F : X \to Y$. The Gateaux derivative of F at $u \subset U$ in the direction φ is defined as*

$$DF(u;\varphi) = \lim_{\epsilon \to 0} \frac{F(u + \epsilon\varphi) - F(u)}{\epsilon}.$$

Theorem A.2 ***(Green's identity)*** *Suppose Ω is an open bounded set of $\Re^N$ and $\partial\Omega$ is a piecewise smooth boundary of Ω. Let u and v are twice continuously differentiable functions, then*

$$\int\limits_{\Omega} \nabla u \nabla v dx = \int\limits_{\partial\Omega} \frac{\partial u}{\partial n} v ds - \int\limits_{\Omega} \nabla^2 u v dx$$

$$\int\limits_{\Omega} c \nabla u \nabla v dx = \int\limits_{\partial\Omega} c \frac{\partial u}{\partial n} v ds - \int\limits_{\Omega} \nabla \cdot (c \nabla u)\, v dx$$

Lemma A.3 ***(Fundamental Lemma of the Calculus of Variations)*** *Let $M(x)$ be a continuous function on the interval $[a, b]$. Suppose that for any continuous function $h(x)$ with $h(a) = h(b) = 0$ we have*

$$\int\limits_{a}^{b} M(x)h(x)dx = 0.$$

Then $M(x)$ is identically zero on $[a, b]$.

Definition A.4 *(Γ-**Convergence**) Let X be a seperable Banach space endowed with a topology τ and let $F_h : X \to \overline{R}$ be a sequence of functionals. We say that F_h Γ-converges to F ($F = \Gamma - lim F_h$) for the topology τ if:*

(i) For every x in X and for every sequence x_n τ-converging to x in X

$$F(x) \leq \underline{lim}_{h\to\infty} F_h(x_h),$$

(ii) For every x in X there exists a sequence τ-converging to x in X such that

$$F(x) \geq \overline{lim}_{h\to\infty} F_h(x_h).$$

Theorem A.5 ***(Integration by parts formula)*** *Ω be an open bounded set of $\Re^N$ and $\partial\Omega$ is C'. Let $u, v \in C'(\overline{\Omega})$, then*

$$\int\limits_{\Omega} u_{x_i} v dx = -\int\limits_{\Omega} u v_{x_i} dx + \int\limits_{\partial\Omega} u v n^i ds \qquad (i = 1, ..., N).$$