ADVANCED NUMERICAL METHODS FOR IMAGE DENOISING AND SEGMENTATION

By

Xiaoyang Liu

A thesis submitted in partial fulfilment of the requirements of

the University of Greenwich

for the Degree of Doctor of Philosophy

Department of Mathematical Sciences

School of Computing and Mathematical Sciences

University of Greenwich

Park Row, London SE10 9LS

2013

DECLARATION

I certify that this work has not been accepted in substance for any degree, and is not concurrently being submitted for any degree other than that of Doctor of Philosophy being studied at the University of Greenwich. I also declare that this work is the result of my own investigation except where otherwise identified by references and that I have not plagiarised the work of others.

Student:

Xiaoyang Liu

Supervisor:

Professor Choi-Hong Lai

Second Supervisor:

Professor Koulis Pericleous

ACKNOWLEDGMENTS

The author wishes to express his gratitude to the University of Greenwich for the sponsorship of this research.

I would like to appreciate my Ph.D. supervisors, Prof. Choi-Hong Lai and Prof. Koulis Pericleous. It is their generous help, support, encouragement, and guidance throughout my Ph.D. study that makes me obtain current achievements. Prof. Lai is a nice, patient person with a very wide knowledge. Working with him not only gives me the academic support, but also expands my scope of knowledge. Prof. Pericleous is a gracious person and willing to give his help to every Ph.D. student in the school of computing and mathematical sciences. It is a great pleasure to work with them these years. I further would like to show my appreciation to Prof. M. Q. Wang from Fuzhou University, China.

I also would like to give my gratitude to other members of CMS. Especially during the CMS seminar, I learnt a lot and many other Ph.D. students from other research areas gave me various suggestions from their point of view. These help me to understand my problems by different angles.

Additionally, I wish to thank all my friends, especially new friends I made in London. It is all of you that give me a colourful life which makes me more concentrating on my study.

Last but not least, I am glad to thank my family. They always support me, help me, encourage me and give me their love. Without them, I would have a very difficult time.

Xiaoyang Liu

London, UK

2013

ABSTRACT

Image denoising is one of the most major steps in current image processing. It is a pre-processing step which aims to remove certain unknown, random noise from an image and obtain an image free of noise for further image processing, such as image segmentation. Image segmentation, as another branch of image processing, plays a significant role in connecting low-level image processing and high-level image processing. Its goal is to segment an image into different parts and extract meaningful information for image analysis and understanding. In recent years, methods based on PDEs and variational functional became very popular in both image denoising and image segmentation. These two branches of methods are presented and investigated in this thesis.

In this thesis, several typical methods based on PDE are reviewed and examined. These include the isotropic diffusion model, the anisotropic diffusion model (the P-M model), the fourth-order PDE model (the Y-K model), and the active contour model in image segmentation. Based on the analysis of behaviours of each model, some improvements are proposed. First, a new coefficient is provided for the P-M model to obtain a well-posed model and reduce the "block effect". Second, a weighted sum operator is used to replace the Laplacian operator in the Y-K model. Such replacement can relieve the creation of the speckles which is brought in by the Y-K model and preserve more details. Third, an adaptive relaxation method with a discontinuity treatment is proposed to improve the numerical solution of the Y-K model. Fourth, an active contour model coupling with the anisotropic diffusion model is proposed to build a noise-resistance segmentation method. Finally, in this thesis, three ways of deriving PDE are developed and summarised. The issue of PSNR is also discussed at the end of the thesis.

Table of contents

List of Figures

List of Tables

Notations and Abbreviations

Notations:

* -- convolution

$C_b^{\infty}(\cdot)$ -- Functions with infinite continuous partial derivatives in bounded space

$C_b(\cdot)$ -- Functions in bounded space

C^m -- m-th order continuous differential function space

$B(x,\varepsilon)$ -- ε-neighbourhood of x

$\langle \cdot,\cdot \rangle$ -- Inner product

$o(\cdot)$ -- Higher order infinitesimal

e -- Identical transformation

D^2 -- Hessian Matrix

∇ -- Gradient operator

∇^2 -- Laplacian operator

div -- Divergence operator

inf -- infimum

sup -- supremum

id -- identical matrix

Abbreviations:

TV model – nonlinear total variation model proposed by Rudin, Osher and Fatemi

P-M model -- the anisotropic diffusion model proposed by Perona and Malik

Y-K model -- the fourth order PDE model proposed by You and Kaveh

IPM model -- the modified coefficient for the anisotropic diffusion model

WYK model -- the weighted sum operator for the fourth order PDE model

AYK model -- the adaptive relaxation method with discontinuity treatment for the fourth order PDE model

PDE-- partial differential equation

MSE -- Mean square error

MAE-- Mean absolute error

PSNR -- Peak signal noise ratio

Chapter 1. **INTRODUCTION**

1.1 Background

Image processing involves various operations applied to an image, such as analysis, enhancement, compression, and reconstruction, to extract some meaningful information or data set related to the image. The techniques developed for image processing in early ages were used to treat analogue signals which are continuous variations of certain quantity with respect to time. In computer science, an analogue signal is first recorded by an optical sensor and then sampled and quantized to generate a digital image.

Digital image processing involves the use of computer to deal with images in order to make them more suitable for image analysis, pattern recognition, and human observation. With the development of computer technology, digital image processing has many applications in different industries. For instance, in traditional engineering fields, photogrammetric software can be used in architectural measurement, casting reconstruction, car manufacturing, and tunnel construction. Other industrial applications include face recognition, steel belt drawbacks detection, traffic monitoring camera, medical imaging. These applications cover areas such as computer vision, remote sensing, medical sciences, weather forecast, intelligent robots, from security to aerospace systems, etc. [1]. It is because of such wide range of applications involving daily life that digital image processing becomes an important research topic. In recent years, many new algorithms have been developed for image processing. These algorithms make use of techniques from different subjects, such as mathematics, signal processing, information science. In addition, methods such as neural network, genetic algorithm, artificial intelligence and fuzzy theory are being explored for image processing. All of these bring significant development in image processing techniques leading to a flourishing research area. Nowadays, image processing includes image restoration, image

segmentation, image edge extraction, shape modelling, feature extraction, object recognition and image visualization amongst others.

Image denoising, as one of the most important steps in image pre-processing, has drawn much interest (see References [1] - [6] and the citations therein). Many other applications in image processing, i.e., image segmentation and edge detection, rely on image denoising as a first step. There are three main directions in image denoising: stochastic modelling, wavelets, and partial differential equations (PDEs). Because of its simple formalism and the extensive numerical analysis literature providing fast, accurate and stable numerical methods, image processing based on PDEs plays an increasing role in real applications. In the literature, the use of PDEs in image processing goes back at least to Gabor [7], and more recently, Jain [8] and many others ([9]-[11]). One typical milestone of the technology is the scale space theory, i.e. representing an image at different scales, proposed in [9] which laid the theoretical foundation for the PDE methods in image processing. Suppose the image domain is Ω, the convolution between an original image u_0 and the Gaussian kernel $G_\sigma(x)$ is given by

$$u(\mathbf{x},\sigma) = u_0(\mathbf{x}) * G_\sigma(\mathbf{x}) = \int_\Omega u_0(\mathbf{y}) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}} d\mathbf{y} . \tag{1.1}$$

Here, u and u_0 correspond to the processed and original images, G_σ is the Gaussian kernel with variance σ. It takes Gaussian convolution to construct a series of images by choosing different values of σ. Koenderink [10] pointed out that (1.1) is equivalent to the solution of the heat equation

$$\frac{\partial u}{\partial t} = \nabla^2 u . \tag{1.2}$$

Image processing by using the heat diffusion in (1.2) is therefore a multi-scale process. This initiated significant work in using PDEs for image processing. However, when using Gaussian convolution or (1.2) to perform denoising, edges of an image cannot be preserved and would lead to edge blurring. Therefore, various models were developed to improve this drawback. One important method proposed by Perona and Malik [12] in 1990 was to employ the magnitude of the gradient, $\|\nabla u\|$, as an edge detector leading to the anisotropic diffusion,

$$\frac{\partial u}{\partial t} = \nabla \cdot (c(\|\nabla u\|)\nabla u) . \tag{1.3}$$

The model defined in (1.3) provides an adaptive diffusion process, i.e. the diffusion is smaller when crossing edges and is larger in areas with small changes in gradient. However, it has been shown that the uniqueness of the solution and stability of the process cannot be guaranteed due to the existence of the backward diffusion problem [13], [20], [50], which leads to an ill-posed problem. Another issue arisen from this model is that it cannot remove noise with large gradient. Attempts to overcome these drawbacks were made by many other researchers. One typical attempt was given by Catté and Coll in [20] where the equation employed a "selective smoothing" of *u*, that is, replacing the gradient by Gaussian gradient of *u* as given below,

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(c\left(\left\| \nabla G_\sigma * u \right\| \right) \nabla u \right), \tag{1.4}$$

to provide a well-posed problem. However, the difficulty of this model is that the choice of a suitable value of σ is not trivial. Small σ may not be able to take effect whereas large σ might lead to loss of features.

Later, Black [14] investigated the impact of the diffusion coefficient $c(\cdot)$ based on an analysis of the discretised form of the anisotropic diffusion and explored the relation between the anisotropic diffusion and robust statistics. Gilboa [15] generalised the diffusion into the complex domain by combing the diffusion equation with the simplified Schrödinger equation. Further work related to the P-M model and other second order PDE models can be found from [16]-[18] and the citations therein.

The development of image denoising also came through the use of variational methods. These methods require the definition of an integral functional usually related to the energy. For example,

$$E(u) = \int_\Omega f(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}), \cdots) d\mathbf{x}, \mathbf{x} \in \Omega, \tag{1.5}$$

may be minimised to obtain an optimal solution for u. One of the most famous models using a variational method is the total variation regularization (TV model) [19] in 1992. It reconstructs u from a noisy image u_0 by defining the energy functional

$$E(u) = \int_\Omega \left\| \nabla u \right\| dxdy + \frac{\lambda}{2} \int_\Omega (u - u_0)^2 dxdy, \tag{1.6}$$

where λ is the Lagrange multiplier. The first term on the right hand side of (1.6) is the regularised term which is used to avoid the instability and the second term is the data fidelity term. Minimising (1.6) leads to the evolution PDE,

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{\|\nabla u\|} \right) + \lambda (u - u_0). \tag{1.7}$$

If λ=0, this model is equivalent to the P-M model by setting $c(s) = 1/s$.

Apart from the second order PDEs, the possibilities of using high order PDEs were also explored by various research peers. Pioneers in this area were You and Kaveh and their fourth order PDE in [23]. The idea is to consider the energy functional

$$E(u) = \int_{\Omega} f(|\nabla^2 u|) dx dy, \tag{1.8}$$

which takes the Laplacian of an image instead of the gradient magnitude into consideration and obtains its minimisation when $|\nabla^2 u| = 0$. The minimisation of (1.8) leads to the fourth order PDE given as below,

$$\frac{\partial u}{\partial t} = -\nabla^2 \left(c(|\nabla^2 u|) \nabla^2 u \right), \tag{1.9}$$

where $c(s) = 1/(1+(s/K)^2)$ and K is a constant. This method brings in some isolated black and white speckles into the denoised image which was pointed out by the authors themselves. Nevertheless, it favours a piecewise harmonic image, i.e. $\nabla^2 u = 0$, as $t \to \infty$, as the approximation and provides more natural images. Hence, it made high order PDEs possible in not only image denoising [24]-[27] but also other image processing [28], [29]. With the above development, PDE-based methods have been widely applied in image edge detection, image segmentation, image inpainting [20]-[22] amongst others.

Although denoising methods based on PDEs have many advantages over other methods, e.g. filtering, there are drawbacks such as the “block effect” in the second order PDE models, which has not been completely remedy, and “isolated speckles” in the fourth-order PDE models. Further exploration is needed to develop enhanced or modified models.

Image segmentation, as another important branch in image processing, is a bridge between low-level processing (including image pre-processing, image processing) and high level processing (image interpretation). The main task of image segmentation is to partition an image into non-intersecting meaningful regions with respect to a particular application. Such meaningful regions could be employed in high-level tasks, i.e., image recognition, object detection. It has a very strong relation with image denoising. The result obtained by image denoising would directly affect the accuracy of image segmentation. It is considered as the first step in object recognition, scene understanding and image understanding [30]. Its applications include industrial quality control, medical image, geophysical exploration, face recognition and so on. In all these applications, the quality of the final results relies reasonably on the pre-processing, including image denoising and segmentation. Mathematically, if Ω is the image domain, image segmentation is to separate Ω into several connected sub-regions $\Omega_i (i = 1, \cdots, n)$ such that

1. $\Omega = \bigcup_{i=1}^{n} \Omega_i$,

2. $\Omega_i \cap \Omega_j = \varnothing, \text{for all } i \text{ and } j, i \neq j$,

3. $p(\Omega_i) = \text{TRUE for all } i$,

4. $p(\Omega_i \cup \Omega_j) = \text{FALSE, for all } i \text{ and } j,\ i \neq j$,

where p is a logical predicate on the points in Ω .

In the literature, methods of image segmentation can be roughly classified into three categories, including boundary-driven segmentation [31], region-driven segmentation [32], and boundary-region-driven segmentation [33]. Variational methods can be applied to all these different categories. The first variational model in the literature was the parametric active contour model (snake model) [32] which was first applied to image segmentation. This model uses an initial closed curve and then slithers the curve towards the salient features of an image (edges, corners, lines and subjective contours). The main idea is to define a curve around the region of interest and obtain the optimal curve by minimising the energy functional

$$J_{snake}(\mathbf{v}(s)) = \int_0^1 E_{int}\left(\mathbf{v}(s)\right) + E_{img}\left(\mathbf{v}(s)\right) + E_{con}\left(\mathbf{v}(s)\right) ds\,, \tag{1.10}$$

with

$$E_{int}(s) = \frac{1}{2}\left(\alpha(s)\left|\mathbf{v}_s(s)\right|^2 + \beta(s)\left|\mathbf{v}_{ss}(s)\right|^2\right). \tag{1.11}$$

Here $\mathbf{v}(s) = (x(s), y(s))$ is the parametric representation of the curve. E_{img} is related to certain property of an image, for instance, the gradient of an image, and E_{con} relates to human operations. The three terms on the right hand side of (1.10) are the internal energy, the image energy, and the constraints. This model describes the deformation of curves and has been applied to various areas related to contour detection [34]-[36] and subsequently many other models were derived including gradient vector flow snake [37], Chan-Vese model [38], and Balloon snake [39].

Another important active contour model is known as the geometric active contour, which was introduced by Caselles [40], [41] and Malladi independently [42]. These models have a strong relationship with curve evolution theory [43] and level set method [44]. Usually, they can be derived by using a Lagrangian formulation that yields a certain evolutionary PDE. As an example, the geodesic contour model defined in [40] is to find a geodesic curve in a Riemannian space, that is,

$$J(C) = \int_0^1 g\left(\left|\nabla u\left(C(q)\right)\right|\right)\left|C'(q)\right| dq\,. \tag{1.12}$$

The corresponding evolutionary PDE can be derived from (1.12) and is given as below,

$$\frac{\partial C(t)}{\partial t} = g(u)\kappa\vec{n} - (\nabla g \cdot \vec{n})\vec{n}\,, \tag{1.13}$$

where g is a stopping function, κ is the Euclidean curvature and $\vec{n}$ is the unit inward normal. Furthermore, if using a level set [44], [45] function ϕ to denote the curve *C*, that is, $C(t) = \{(x, y) \mid \phi(x, y, t) = 0, t \in R^+\}$, the corresponding level set form of (1.13) can be written as

$$\frac{\partial \phi}{\partial t} = g(u)\left|\nabla \phi\right|\kappa + \nabla g(u) \cdot \nabla \phi\,. \tag{1.14}$$

This approach has several advantages over the traditional parametric active contours model, such as the change of topology, i.e. the curve can split itself when meeting several desired objects. However, the level set function is computed in a higher dimension and it is numerically necessary to keep the level set function close to a signed distance function [46], which makes the cost of computation quite high. Fortunately, this problem was solved by Li et al. [47] and their following research [48]. There they defined a new integral to force the level set function to be close to a signed distance function and therefore avoided the costly re-initialisation procedure.

1.2 Aims of the Study

The aim of this thesis is to examine improved PDE models for image processing, as well as their numerical implementations with socio-economic applications such as heritage, forensic and CCTV images. Mathematical formulations using variational methods leading to several PDE models for image segmentation are also studied and analysed. Advanced numerical methods have been developed during the course of investigation. In detail, during the study, the following objectives are expected to be achieved:

1. Explore the concept of scale space theory and its relationship to PDE methods in image denoising.
2. Investigate PDE models and their drawbacks in image denoising.
3. Examine improved models to overcome various drawbacks of PDE methods.
4. Explore the use of the variational method in image processing and its application in image denoising and segmentation.
5. Examine various image segmentation models based on active contour models and curve evolution.
6. Integrate the anisotropic diffusion model into segmentation models to build a noise-resistance image segmentation model.

1.3 Contribution of this Study

There are several important contributions of this study. These contributions are summarised below.

1. Summarise and develop several ways of building PDEs for image processing. The study exposes the relationship between PDE methods and other methods in image processing and presents the validity of using PDE methods for image processing.
2. With the analysis of behaviours of P-M model and the reason of "block effect", a new coefficient is proposed to avoid the backward diffusion. This coefficient not only achieves a well-posed equation, but also relieves the "block effect" drawback in the P-M model as shown by the experiments.
3. Choices of the value of the threshold in the Y-K model is discussed and explored. Aiming to avoid the speckles brought in the Y-K model, a weighted sum operator is proposed to replace the Laplacian which can relieve the generation of speckles, based on the investigation of the reason of speckles.
4. From the view of numerical implementation, three alternations, including an adaptive relaxation method, discontinuity detection, and restoration is employed to remove the speckles which are brought in by Y-K model.
5. A stopping criterion is proposed which can be used to make the iteration stopping automatically for the fourth order PDE.
6. Behaviours of active contour model and discussion about the choice of coefficients of the internal energy are investigated. The anisotropic diffusion model is integrated into the AC model to create a noise-resistance segmentation model.

1.4 Structure of the thesis

This thesis consists of chapters as following:

Chapter 1 gives a brief background introduction and aims of this study as well as the new contributions which the author acheived.

Chapter 2 provides some related mathematical background for the study, including the scale space theory and variational methods. One is the theoretical foundation of the PDE methods in image processing and the other is an important tool for modelling image processing problems.

Chapter 3 reviews the basic background of image processing, which includes image modelling, image noise and summarises the methodologies in image processing. At the end

of this chapter, the relation between PDE and image processing is discussed and the advantage of using PDE methods in image processing is addressed.

Chapter 4 focuses on some typical PDE models and analyses their behaviours as well as their advantages and drawbacks. Numerical implementation for each model is provided at the end of each section.

Chapter 5 describes some improved PDE models proposed by the author. These include IPM, WYK, and AYK models. Motivations of each model are addressed as well as the implementation details. A stopping criterion also is given at the end of this chapter.

Chapter 6 is to analyse the behaviour of active contour model with some demonstrations and then introduce the anisotropic diffusion into the active contour model to build a noise-resistance segmentation model. The choices of the coefficient in the P-M model and its effects on the segmentation result are discussed.

Chapter 7 states and compares the numerical results of each denoising model examined in this thesis and extends the discussion to the measurement of image quality.

Chapter 8 summarises the conclusion of this work and address the future research possibilities.

Chapter 2. MATHEMATICAL PRELIMINARIES

This chapter focuses on introducing various mathematical notions that form the basic foundation or are used in this thesis. The first part of this chapter concerns scale space theory. This lays the theoretical foundation for PDE methods in image processing. The second part relates to the calculus of variation, in which the principle of variational methods and various formulations using energy functional are discussed.

2.1 Scale Space Theory

Objects of the world usually have a hierarchical structure which can be considered as meaningful entity by using different ranges of scales. For example, at the finest scale, a tree consists of leaves with branches, whereas at the coarsest scale, a tree may be composed of several limbs. Similar examples can also be observed in geographical maps. A map of the world can only include countries and major cities, possibly, whereas a map of a city may show the towns and even some well-known boulevards. In physics, objects are modelled at different levels of scales as well, ranging from quantum at fine scales to celestial bodies at coarse scales. The concept of the scale space was initially introduced by Iijima [49] in 1962, but it was only in 1983 when Witkin [9] discussed the smoothing of signals, that the scale space theory was applied to computer vision and the entire theory was made mature by Alvarez [50] and Lindeberg [51], [52] in 1994. More information about scale space theory and its history can be found in references [53]-[57] and the citations therein.

2.1.1 Definition of the Scale Space

Scale space theory is an abstract framework used to handle images with multi-scale representation. The main idea is to obtain a family of derived images at different scales by convolving the original image with a one-parameter Gaussian kernel, from which suitable

scale can be chosen for specific applications. In order to address this theory, some basic definitions related to the scale space are given below.

Definition 2.1 Let U be a space of functions on $\Omega \subset \mathbb{R}^n$ and $T_t : U \to U$ is a family of operators on U, where $t \geq 0$, denoted by $\{T_t\}_{t \in R_+}$. $\{T_t\}_{t \in R_+}$ is a scale space if $u(t,x) = (T_t u_0)(x)$, where $u_0 \in U$.

■

In this definition, t is a scale parameter which is a measurement of the smoothing effect and $T_t u_0$ is the mapping function of u_0 at the scale t. T_t is known as a scale space operator. In image processing, u_0 typically represents a given image that needs to be processed.

2.1.2 Structure of the Scale Space

Definition 2.2 For a scale space operator T_t, $\forall t, h \geq 0$, if there exists a family of transition operators $\left\{T_{t+h}^t\right\}$, such that

$$T_0 = e,$$

$$T_{t+h} = T_{t+h}^t T_t,$$

then T_t has a pyramidal structure (as illustrated in Figure 2.1).

■

Figure 2.2 depicts such a pyramidal structure. In a pyramidal scale space, the more smoothing function $T_t u_0$ can be calculated from the function $T_s u_0$ ($0 < s < t$), without any prior knowledge of $T_{s'} u_0$ ($0 \leq s' < s$). This means that images defined at larger scales can be directly derived from images defined at smaller scales under the scale space operator.

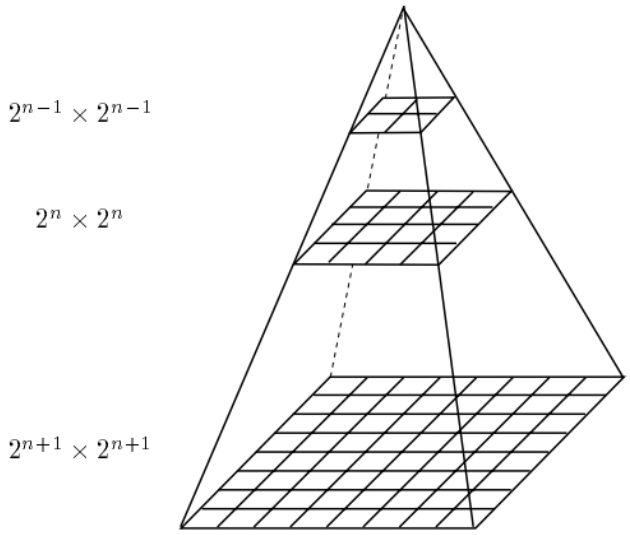


Figure 2.1: A pyramid presentation.

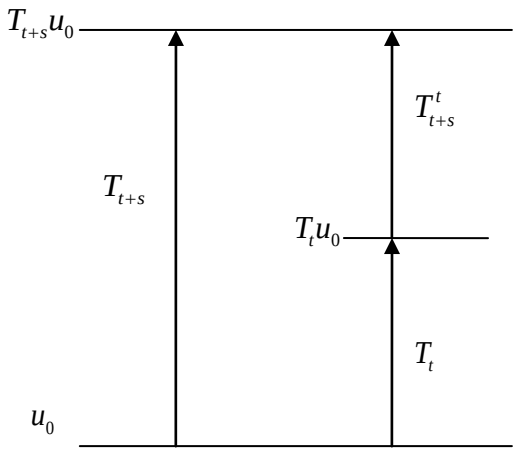


Figure 2.2: Scale space.

2.1.3 Axioms of the Scale Space

In the next two sections, several axioms for the scale space are introduced. Amongst these properties, some are used to deal with architectural properties of the multi-scale image analysis (such as recursivity, regularity, locality, and causality) whereas the others (invariance) are related to the property of shape-preserving when analysing the scale space using partial differential equations (PDEs) [50] - [56].

Suppose $\{T_t\}_{t\in R_+}$ is a scale space, from Definition 2.1, without imposing the well-defined structure as in Definition 2. 2, the properties below hold.

- **Recursivity**: $\{T_t\}_{t\in R_+}$ is **recursive** if $T_0 = e$ and $\forall s,t \geq 0$, $T_{t+s} = T_t \circ T_s$.

■

This property is also known as the **semi-group property**. It means that signals at coarse scales can be derived by iterating signals at fine scales without other information and the greater the scale parameter t is, the simpler the signal content is.

- **Regularity**: Suppose $x \in \Omega, y \in B(x,\varepsilon)$ and *u* has a quadratic expression, i.e.,

$$u(y) = \frac{1}{2}\langle A(y-x), y-x\rangle + \langle p, y-x\rangle + c, \tag{2.1}$$

where A is a symmetric $n\times n$ matrix, $p \in \mathbb{R}^n$ and $c \in \mathbb{R}$. If there exists a function $F(A, p, x, c, t)$ such that it is continuous with respect to *A* and

$$\lim_{h\to 0} \frac{(T_{t+h}u - u)(x)}{h} \to F(A, p, x, c, t), \tag{2.2}$$

then $\{T_t\}_{t\in R_+}$ is **regular**.

■

The operator T_t and the scale space are continuous if the regularity property holds. Therefore, it guarantees that the signal is smoothed continuously as *t* increases.

In the case of image processing, by using Taylor's theorem, it is trivial to find out that $c = u(x), p = \nabla u(x), A = D^2u(x)$.

- **Local Comparison Principle**: Given that $x \in \Omega$, $y \in \mathrm{B}(x,\varepsilon)$, if $\forall u, v \in X$ and $t \geq 0$, $u(y) \leq v(y)$, then there exists a sufficiently small *h*, subject to $(T_{t+h}^t u)(x) \leq (T_{t+h}^t v)(x)$. If this holds for an arbitrary point in Ω, then $\{T_t\}_{t\in R_+}$ satisfies the **local comparison principle**.

■

The local comparison principle ensures that a scale space does not introduce any new features to the data as scale *t* increases.

- **Causality**: $\{T_t\}_{t\in R_+}$ is **causal** if it is regular and meets the pyramidal structure and the local comparison principle at the same time.

■

Causality ensures that the local extrema could not be enhanced by using scale space operators.

With these axioms, the relation between the scale space and PDEs can be built by the following theorem [50]:

Theorem 2.1: If $u\in C^2$ and a scale space $\{T_t\}_{t\in R_+}$ is causal, then there exists a function $F(A,p,x,c,t)$ satisfies

$$\frac{(T_{t+h}u-u)(x,t)}{h}\to F(D^2u(x),Du(x),u(x),x,t),as\ h\to 0, \tag{2.3}$$

where F is also known as the correlation function of the scale space. Furthermore, $F(D^2u(x),Du(x),u(x),x,t)$ is a non-decreasing function with respect to D^2u. In other words, for two symmetric matrices, A and $\tilde{A}$, if $A-\tilde{A}$ is a positive definite matrix, then

$$F(A,p,x,c,t)\ge F(\tilde{A},p,x,c,t), \tag{2.4}$$

Proof:

Let $v(x)=\frac{1}{2}\langle Ax,x\rangle+\langle p,x\rangle+c$ be a quadratic form in $\mathbb{R}^2$. According to the definition of regularity,

$$\lim_{h\to 0^+}\frac{(T_{t+h}v-v)(x)}{h}\to F(A,p,x,c,t). \tag{2.5}$$

The function F is thus continuous with respect to the first parameter A of the correlation function. Suppose $u:\mathbb{R}^2\to\mathbb{R}$ and $u\in C^2$ at a point $x\in\Omega$, it is required to prove that (2.3) holds for u at the point x and function F in (2.3) and (2.5) are the same functions. Since $u\in C^2$ at the point x, for $y\in B(x,\varepsilon)$,

$$u(y)=u(x)+\langle Du,y-x\rangle+\frac{1}{2}\langle(D^2u)(y-x),y-x\rangle+o\left(|x-y|^2\right), \tag{2.6}$$

where Du and D^2u are the first and second order differential operators, respectively. $\forall\eta>0$, define the following quadratic forms:

$$Q^+(y)=u(x)+\langle Du,y-x\rangle+\frac{1}{2}\langle D^2u(y-x),y-x\rangle+\frac{1}{2}\eta(x-y)^2, \tag{2.7}$$

$$Q^-(y)=u(x)+\langle Du,y-x\rangle+\frac{1}{2}\langle D^2u(y-x),y-x\rangle-\frac{1}{2}\eta(x-y)^2. \tag{2.8}$$

When $y\neq x$ and if $|y-x|$ is sufficiently small,

$$Q^-(y)<u(y)<Q^+(y), \tag{2.9}$$

According to the local comparison principle,

$$(T^t_{t+h}Q^-)(x)\le(T^t_{t+h}u)(x)\le(T^t_{t+h}Q^+)(x). \tag{2.10}$$

From (2.7) and (2.8), it can be easily seen that:

$$Q^-(x)=u(x)=Q^+(x). \tag{2.11}$$

So

$$(T^t_{t+h}Q^-)(x)-Q^-(x)\le(T^t_{t+h}u)(x)-u(x)\le(T^t_{t+h}Q^+)(x)-Q^+(x). \tag{2.12}$$

Divide (2.12) by *h*, and let $h\to0^+$. Since $\{T_t\}_{t\in R_+}$ is regular, one obtains

$$\begin{aligned}
&\lim_{h\to0^+}\frac{(T^t_{t+h}Q^-)(x)-Q^-(x)}{h}\\
&\le\lim_{h\to0^+}\inf\frac{(T^t_{t+h}u)(x)-u(x)}{h}\\
&\le\lim_{h\to0^+}\sup\frac{(T^t_{t+h}u)(x)-u(x)}{h}\\
&\le\lim_{h\to0^+}\frac{(T^t_{t+h}Q^+)(x)-Q^+(x)}{h}.
\end{aligned}$$

Hence

$$\begin{aligned}
&F(D^2u-\eta\times id,Du,u,x,t)\\
&\le\lim_{h\to0^+}\inf\frac{(T^t_{t+h}u)(x)-u(x)}{h}\\
&\le\lim_{h\to0^+}\sup\frac{(T^t_{t+h}u)(x)-u(x)}{h}\\
&\le F(D^2u+\eta\times id,Du,u,x,t).
\end{aligned}$$

As $\{T_t\}_{t\in R_+}$ is regular, so when $\eta\to0^+$,

$$\lim_{h\to 0^+}\frac{(T^t_{t+h}u)(x)-u(x)}{h}=F(D^2u,Du,u,x,t). \tag{2.13}$$

It is required to prove that F is non-decreasing with respect to *A*.

For an arbitrary vector *p* and any matrices $A,\tilde{A}$, such that $A\geq\tilde{A}$, let

$$\alpha^+(y)=c+\langle p,y-x\rangle+\frac{1}{2}\langle A(y-x),y-x\rangle+\frac{1}{2}\eta(x-y)^2, \tag{2.14}$$

$$\alpha^-(y)=c+\langle p,y-x\rangle+\frac{1}{2}\langle \tilde{A}(y-x),y-x\rangle-\frac{1}{2}\eta(x-y)^2. \tag{2.15}$$

Then in certain neighbourhood of *x*,

$$\alpha^-(y)<\alpha^+(y). \tag{2.16}$$

By using local comparison,

$$(T^t_{t+h}\alpha^-)(x)<(T^t_{t+h}\alpha^+)(x), \tag{2.17}$$

so

$$\lim_{h\to 0^+}\frac{(T^t_{t+h}\alpha^-)(x)-\alpha^-(x)}{h}\leq\lim_{h\to 0^+}\frac{(T^t_{t+h}\alpha^+)(x)-\alpha^+(x)}{h}, \tag{2.18}$$

that is,

$$F(\tilde{A}-\eta\times id,p,x,c,t)\leq F(A+\eta\times id,p,x,c,t). \tag{2.19}$$

Let $\eta\to 0^+$, one obtains

$$F(A,p,x,c,t)\geq F(\tilde{A},p,x,c,t). \tag{2.20}$$

■

Theorem 2.1 is an important result that creates the mapping between a scale space and a partial differential equation. In other words, given a scale space T_t, the corresponding parabolic PDE can be defined by using Theorem 2.1 and vice versa. Given a parabolic PDE with a solution $u(x,t)$, then according to Definition 2.1, the scale space operator can be defined by $T_t:u_0(x)\to u(x,t)$, where $u_0(x)$ is the initial function. Note that the scale

parameter *t* in scale space is the same as the time parameter *t* in the time-dependent PDEs. This explains the reason of considering *t* as the scale parameter in the scale space setup.

2.1.4 Scale Space and Invariance

In the real world, pictures of the same scene may be visually different due to the influence of the shooting positions and environments. As a result, special requirements are needed for image filtering. With this in mind, different forms of invariance are required for different scale spaces.

- **Grey-scale-shift invariance (linearity)**: A scale space $\{T_t\}_{t\in R_+}$ is **grey-scale-shift invariant** if $\forall u \in U$ and a constant *c*,

$$T_{t+h}^t(0) = 0,$$

$$T_{t+h}^t(u+c) = T_{t+h}^t(u) + c\,.$$

Theorem 2.2: If a scale space $\{T_t\}_{t\in R_+}$ is causal, and is grey-scale-shift invariant, then the correlation function $F(A, p, x, c, t)$ is independent on *c*.

■

- **Translation invariance**: A scale space $\{T_t\}_{t\in R_+}$ is **translation invariant if** $\forall y, t \geq 0, h \geq 0$

$$T_{t+h}^t(\tau_y \cdot u) = \tau_y \cdot (T_{t+h}^t u),$$

where $\tau_y \cdot u(x) = u(x-y)$.

Theorem 2.3: If a scale space $\{T_t\}_{t\in R_+}$ is causal, and is translation invariant, then the correlation function $F(A, p, x, c, t)$ is independent on *x*.

■

- **Euclidean invariance**: A scale space is **Euclidean invariant** if for an arbitrary isometric transformation *D*,

$$DT_{t+h}^t = T_{t+h}^t D\,,$$

where $Du(x) = u(Dx)$.

Theorem 2.4: If a scale space is causal, grey-scale-shift invariant and Euclidean invariant, its correlation function *F* satisfies:

$$F(DAD^T, Dp, t) = F(A, p, t), \forall D. \tag{2.21}$$

■

- **Morphological invariance**: A scale space $\{T_t\}_{t\in R_+}$ is **morphologically invariant** if for any increasing function $g:[0,1]\to[0,1]$, it satisfies

$$g \circ T_{t+h}^t = T_{t+h}^t \circ g .$$

- **Scale invariance**: A scale space $\{T_t\}_{t\in R_+}$ is **scale invariant**, if $\forall \lambda, t > 0$, there exists $t'(t,\lambda) > 0$ such that

$$H_\lambda T_{t'}^{s'} = T_t^s H_\lambda ,$$

where $H_\lambda(u) = u(\lambda x)$ and $t' = t'(t,\lambda), s' = s'(t,\lambda)$.

- **Affine invariance**: A scale space is **affine invariant**, if for any affine transformation *B*, there exists $t'(t,B) > 0$ such that

$$BT_{t'}^{s'} = T_t^s B,$$

where $t' = t'(t,B), s' = s'(t,B)$.

Using the above scale space theories, an example showing the connection between scale space and PDE is given below. The proof of this connection is given in Chapter 3.

Example 2.1 Given a scale space $\{T_t\}$ defined by $T_t u = u(t)$, if it is casual, translation and Euclidean invariant, then $u(t)$ is the solution of the heat equation

$$\begin{cases} \dfrac{\partial u(t)}{\partial t} = \nabla^2 u(t) \quad (\mathbf{x}, t) \in \Omega \times (0, T] \\ u(0) = u_0 \end{cases} \tag{2.22}$$

In other words, a linear and casual scale space with translation and Euclidean invariance is equivalent to the heat equation. This type of scale space is a linear scale space or known as

Gaussian scale space. Other types of scale space have their corresponding PDEs, for instance, the PDE corresponding to the affine morphological scale space is

$$\frac{\partial u}{\partial t} = |Du|\left(t\ \kappa(u)\right)^{\frac{1}{3}}. \tag{2.23}$$

More information about different types of scale space and their PDE can be found in [58], [59].

2.2 Variational Methods

The variational method is based on the calculus of variations and is a mathematical method that involves finding an optimal solution from maximising or minimising the corresponding functional. The history of variational methods can be dated back to the seventeenth century when some functional extreme-value problems were proposed associated with the research of geometry and theoretical mechanics which led to the development of the theory and practical applications of the variational method [60]-[63]. In the past few years, the variational method is also widely used in image processing, including image restoration, image segmentation, shape from shading [21], [23], [64]. In this section, a brief introduction is given of the calculus of variations related to image processing.

2.2.1 Fundamental Problem

The fundamental problem of the calculus of variations is as follows. If *a*, *b* are two fixed real numbers, $b > a$, and the function $F(x,u,u')$ is a given function of the three independent variables x,u,u', the problem is to find amongst the admissible functions the particular one, $u = u(x)$, that minimises the integral

$$J = \int_a^b F(x,u,u')dt\,. \tag{2.24}$$

2.2.2 Basic Definitions

The definitions below assist the description in later sections and chapters.

Definition 2.3 Let *S* be a set of functions. If for any function $u(x) \in S$, there exists a real value $J \in \mathbb{R}$ corresponding to it, then *J* is called as a **functional** on *S*, denoted by $J(u(x))$.

Here, S is also known as a set of admissible functions. For simplicity, a functional can be considered as a generalized function with respect to functions.

■

Definition 2.4 Suppose two functions $u(x), u_0(x) \in C^n(\Omega)$, Ω is the domain of all functions, then the **n-th order distance** between these two functions is defined as: $\rho_n(x, x_0) = \max\limits_{0 \le i \le n} \max\limits_{x \in \Omega} \left\{ \left| u^{(i)}(x) - u_0^{(i)}(x) \right| \right\}$. Especially, when $n = 1$, the notation can be simplified as $\rho(u, u_0)$.

■

Definition 2.5 Suppose $u_0(x) \in C^n(\Omega)$, then **the n-th order neighbourhood** of $u_0(x)$ is defined as: $B_n(u_0(x), \varepsilon) = \left\{ u(x) \mid u(x) \in C^n(\Omega), \rho_n(u(x), u_0(x)) < \varepsilon \right\}$.

■

Definition 2.6 (a) Suppose $u_0 \in S$, the **variation of a function** on u_0 is written δu and defined as $\delta u = u - u_0, \forall u \in S, u \ne u_0$. Denote the increment of functional by $\Delta J = J(u_0 + \delta u) - J(u_0)$. If this increment can be written as $\Delta J = L(u_0, \delta u) + o(\rho)$, where the first operator L is linear with respect to δu and ρ is the distance between u_0 and u, then L is called as the **first variation of a functional**, denoted by δJ.

■

There is one equivalent definition as given below:

Definition 2.6 (b) For a functional $J(u(x))$, there exists a function $\Psi(\varepsilon) = J[u(x) + \varepsilon \delta u]$, if $\Psi'(0) = \left. \dfrac{\partial J[u(x) + \varepsilon \delta u]}{\partial \varepsilon} \right|_{\varepsilon = 0}$ exists, then $\Psi'(0)$ is the **first variation of the functional** $J(u(x))$, denoted by δJ.

■

Definition 2.7 A functional $J(u(x))$ has an **absolute minimum (maximum) value** at $u_0(x) \in S$ means that for any $u(x) \in S$, then $J(u(x)) \ge J(u_0(x))$ ($J(u(x)) \le J(u_0(x))$).

■

Definition 2.8 A functional $J(u(x))$ has a **strong relative minimum (maximum) value** at $u_0(x) \in S$ means that for any $u(x) \in B_0(u_0(x),\ \varepsilon) \cap S,\ \varepsilon > 0$, then $J(u(x)) \geq J(u_0(x))$ ($J(u(x)) \leq J(u_0(x))$).

■

Definition 2.9 A functional $J(u(x))$ has a **weak relative minimum (maximum) value** at $u_0(x) \in S$ means that for any $u(x) \in B_1(u_0(x), \varepsilon) \cap S, \varepsilon > 0$, then $J(u(x)) \geq J(u_0(x))$ ($J(u(x)) \leq J(u_0(x))$).

■

2.2.3 Necessary Conditions for a Weak Relative Minimum

In order to establish the fundamental theorem of the calculus of variations, the lemma below is needed and the necessary conditions for a weak relative minimum for (2.24) is given afterwards.

Lemma 2.1 If a function $f(x) \in C[a,b]$ and for any function $\eta(x) \in C^2[a,b]$ with $\eta(a) = \eta(b) = 0$, and if

$$\int_a^b f(x)\eta(x)dx = 0, \tag{2.25}$$

then $f(x) = 0$.

■

Suppose the function $\Gamma_0 : u = u(x)$ is the required function which minimises (2.24) satisfying the boundary condition

$$u(a) = u_0, u(b) = u_1. \tag{2.26}$$

Now consider the varied function defined by

$$u = u(x) + \alpha\eta(x), \tag{2.27}$$

where $\eta(x) \in C^2[a,b]$ and $\eta(a) = \eta(b) = 0$. Denote Γ_α as the one-parameter family of functions given by (2.27). Let $\psi(\alpha) = J(\Gamma_\alpha)$ such that $\psi(\alpha)$ is a function of the single variable α. For Γ_0 to be a minimum of (2.24), the necessary condition is

$$\psi'(0) = 0. \tag{2.28}$$

Since

$$\begin{aligned}\psi(\alpha) &= \int_a^b F\left(x, u(x) + \alpha\eta(x), u'(x) + \alpha\eta'(x)\right) dx \\ &= \int_a^b F(x, \Phi, \Phi') dx,\end{aligned}$$

where $\Phi = u(x) + \alpha\eta(x), \Phi' = u'(x) + \alpha\eta'(x).$

$$\begin{aligned}\psi'(\alpha) &= \int_a^b \frac{\partial F}{\partial \alpha} dx \\ &= \int_a^b \left(\frac{\partial F}{\partial \Phi}\frac{\partial \Phi}{\partial \alpha} + \frac{\partial F}{\partial \Phi'}\frac{\partial \Phi'}{\partial \alpha} \right) dx \\ &= \int_a^b \left(\eta(x)\frac{\partial F}{\partial \Phi} + \eta'(x)\frac{\partial F}{\partial \Phi'} \right) dx.\end{aligned}$$

Putting $\alpha = 0$ leads to $\Phi = u(x), \Phi' = u'(x)$, and the necessary condition becomes

$$\psi'(0) = \int_a^b \left(\eta(x)\frac{\partial F}{\partial u} + \eta'(x)\frac{\partial F}{\partial u'} \right) dx = 0.$$

Integrating the second term by parts gives

$$\psi'(0) = \int_a^b \left(\eta(x)\frac{\partial F}{\partial u} - \eta(x)\frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right) \right) dx + \left[\eta(x)\frac{\partial F}{\partial u'} \right]_a^b. \tag{2.29}$$

Using the boundary conditions $\eta(a) = \eta(b) = 0$, the last term vanishes. Hence the necessary condition becomes

$$\psi'(\alpha) = \int_a^b \eta(x)\left(\frac{\partial F}{\partial u} - \frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right) \right) dx. \tag{2.30}$$

(2.30) holds for all functions $\eta(x) \in C^2[a,b]$ with $\eta(a) = \eta(b) = 0$. Using Lemma 2.1 leads to

$$\frac{\partial F}{\partial u} - \frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right) = 0. \tag{2.31}$$

■

This fundamental result shows the necessary condition for the function $u = u(x)$ to exhibit a weak relative minimum for the functional in defined in (2.24). It is known as **Euler's equation** (or sometimes Euler-Lagrange equation). The functions or curves defined by (2.31), i.e. Γ_0, are called **extremals** for the functional $J = \int_a^b F(x,u,u')dt$.

2.2.4 Euler's Equation

In the above section, the necessary condition for a weak relative minimum for a functional that is, the Euler's equation, is discussed. In this section, some special but common cases, of Euler's equation are briefly discussed.

- F is independent of u', one has $F = F(x,u)$ or $F = F(u)$, then $F_{u'} \equiv 0$. (2.31) becomes $F_u(x,u) = 0$ or $F_u(u) = 0$. In most cases, the solution of both equations cannot satisfy the boundary conditions, so there is no solution for the variational problem.
- F is independent of u, one has $F = F(x,u')$. In such case, the Euler's equation is $\frac{d}{dx}F_{u'}(x,u') = 0$. Thus the solution is a one-parameter family of functions $u = \int_a^b \varphi(x,c)dx$, where c is a constant.
- F is only dependent of u', one has $F = F(u')$. Then the Euler's equation becomes $u''F_{u'u'} = 0$. Two different cases arise. When $u'' = 0$, the solution is $u = c_1 x + c_2$, which is a two-parameters family of straight line. When $F_{u'u'} = 0$, it is possible to assume n real roots, i.e. $u' = r_i (i = 1,2,\cdots,n)$, leading to the solutions $u = r_i x + C(i = 1,2,\cdots,n)$. Consequently, the solution for the variational problem is a family of straight line $u = c_1 x + c_2$.

- F is independent of x explicitly. Hence $F = F(u, u')$ and the Euler's equation becomes $F_u - F_{uu'} - F_{u'u'}u'' = 0$. By integration, the solution is $x = \int \frac{du}{\varphi(u, c_1)} + c_2$, where $u' = \varphi(u, c_1)$ and c_1, c_2 are two constants.

In real applications, the functional would depend on several functions instead of one function, or multiple unknown variables instead of one unknown variable, or higher order derivatives instead of the first derivative. For example, in image processing, higher dimension such as 3-dimension can be found in 3D image construction. In the section follows, several generalized cases of variational problems are introduced.

Theorem 2.5 (**Necessary conditions for a weak relative minimum for vaiational problems dependent on two functions**): If the functions $u = u(x)$ and $v = v(x)$ minimise the functional

$$J[u(x), v(x)] = \int_{x_0}^{x_1} F(x, u, u', v, v')dx, \tag{2.32}$$

with the boundary conditions

$$u(x_0) = u_0, u(x_1) = u_1, v(x_0) = v_0, v(x_1) = v_1,$$

then they satisfy the system of Euler's equations

$$\begin{cases} \frac{\partial F}{\partial u} - \frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right) = 0 \\ \frac{\partial F}{\partial v} - \frac{d}{dx}\left(\frac{\partial F}{\partial v'}\right) = 0 \end{cases}. \tag{2.33}$$

■

Corollary 2.1 If the set of functions $u_1(x), u_2(x), \cdots, u_n(x)$, minimises of the functional

$$J(u_1, u_2, \cdots, u_n) = \int_{x_0}^{x_1} F(x, u_1, u_2, \cdots, u_n, u_1', u_2', \cdots u_n')dx, \tag{2.34}$$

with the boundary conditions

$$u_i(x_0) = u_{i0}, u_i(x_1) = u_{i1},$$

then they satisfy the system of Euler's equations

$$\frac{\partial F}{\partial u_i} - \frac{d}{dx}\left(\frac{\partial F}{\partial u_i^{'}}\right) = 0, i = 1, 2, \cdots, n . \tag{2.35}$$

■

Theorem 2.6 (Necessary conditions for a weak relative minimum for vaiational problems dependent on higher order derivatives) Suppose two functions $u(x) \in C^4[x_0, x_1]$ and $F(x, u, u', u'') \in C^3[x_0, x_1]$, then the necessary condition to minimize the functional

$$J[u(x)] = \int_{x_0}^{x_1} F(x, u, u', u'')dx , \tag{2.36}$$

with the boundary conditions

$$u(x_0) = u_0, u(x_1) = u_1, u'(x_0) = u_0^{'}, u'(x_1) = u_1^{'},$$

is that the extremal function $u = u(x)$ satisfies

$$\frac{dF}{du} - \frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right) + \frac{d^2}{dx^2}\left(\frac{\partial^2 F}{\partial u''}\right) = 0 . \tag{2.37}$$

■

(2.37) is usually known as **Euler-Poisson's equation** or **Euler's equation**. Theorem 2.6 can also be generalized to variational problems including higher order derivatives, or more than one variable, which are described in Corollary 2.2.

Corollary 2.2 (a): The necessary condition that a function $u(x) \in C^{2n}[x_0, x_1]$ minimises the functional

$$J(u) = \int_{x_0}^{x_1} F(x, u, u', \cdots, u^{(n)})dx , \tag{2.38}$$

with the boundary conditions

$$u^{(k)}(x_0) = u_0^{(k)}, u^{(k)}(x_1) = u_1^{(k)}, k = 0, 1, \cdots, n-1,$$

is that the extreme curve $u = u(x)$ satisfies the Euler-Poisson's equation

$$\frac{\partial F}{du}-\frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right)+\frac{d^2}{dx^2}\left(\frac{\partial F}{\partial u''}\right)-\cdots+(-1)^n\frac{d^n}{dx^n}\left(\frac{\partial F}{\partial u^n}\right)=0, \tag{2.39}$$

where $F\in C^{n+1}[x_0,x_1]$.

■

For convenience, denote $F_u^{(k)}=\frac{\partial F}{du^k},k=0,1,\cdots,n$ and if $k=0,F_u^{(0)}=F_u$, then (2.39) can be simplified as

$$\sum_{k=0}^{n}(-1)^k\frac{d^k}{dx^k}F_{u^{(k)}}=0. \tag{2.40}$$

Corollary 2.2 (b) The necessary condition of minimising the functional

$$J(u,v)=\int_{x_0}^{x_1}F(x,u,u',\cdots,u^{(m)},v,v',\cdots,v^{(n)})dx, \tag{2.41}$$

which depends on two unknown functions, $u(x)\in C^m[x^0,x^1]$, $v(x)\in C^n[x^0,x^1]$, with the boundary conditions

$$u^{(k)}(x_0)=u_0^{(k)},u^{(k)}(x_1)=u_1^{(k)}\quad(k=0,1,\cdots,m-1),$$

$$v^{(k)}(x_0)=v_0^{(k)},v^{(k)}(x_1)=v_1^{(k)}\quad(k=0,1,\cdots,n-1),$$

is that the extreme curves $u=u(x),v=v(x)$ satisfy the system of Euler's equation

$$\begin{cases}\dfrac{\partial F}{du}-\dfrac{d}{dx}\left(\dfrac{\partial F}{\partial u'}\right)+\dfrac{d^2}{dx^2}\left(\dfrac{\partial F}{\partial u''}\right)-\cdots+(-1)^n\dfrac{d^m}{dx^m}\left(\dfrac{\partial F}{\partial u^m}\right)=0\\ \dfrac{\partial F}{dv}-\dfrac{d}{dx}\left(\dfrac{\partial F}{\partial v'}\right)+\dfrac{d^2}{dx^2}\left(\dfrac{\partial F}{\partial v''}\right)-\cdots+(-1)^n\dfrac{d^n}{dx^n}\left(\dfrac{\partial F}{\partial v^n}\right)=0\end{cases}. \tag{2.42}$$

Or

$$\begin{cases}\displaystyle\sum_{k=0}^{m}(-1)^k\frac{d^k}{dx^k}F_{u^{(k)}}=0\\ \displaystyle\sum_{k=0}^{n}(-1)^k\frac{d^k}{dx^k}F_{v^{(k)}}=0\end{cases}. \tag{2.43}$$

■

Corollary 2.2 (c) The necessary condition of minimising the functional

$$J(u_1,u_2,\cdots,u_m)=\int_{x_0}^{x_1}F(x,u_1,u_1^{'},\cdots,u_1^{(n_1)},u_2,u_2^{'},\cdots,u_2^{(n_2)},\cdots,u_m,u_m^{'},\cdots,u_m^{(n_m)})dx\,, \quad (2.44)$$

which depends on unknown functions $u_i(x)\in C^{n_i}[x_0,x_1]$ $(i=1,2,\cdots,m)$ with the boundary conditions

$$u_i^{(k)}(x_0)=u_{i0}^{(k)},u_i^{(k)}(x_1)=u_{i1}^{(k)} \quad (i=1,2,\cdots,m,k=0,1,\cdots,n_i-1)\,,$$

is that the extreme curves $u_i=u_i(x)$ $(i=1,2,\cdots,m)$ satisfy Euler's equation

$$F_{u_i}-\frac{d}{dx}F_{u_i^{'}}+\frac{d^2}{dx^2}F_{u_i^{''}}-\cdots+(-1)^{n_i}\frac{d^{n_i}}{dx^{n_i}}F_{u_i^{(n_i)}}=0\,. \quad (2.45)$$

Or

$$\sum_{k=0}^{m}(-1)^k\frac{d^k}{dx^k}F_{u_i^{(k)}}=0,i=1,2,\cdots m. \quad (2.46)$$

■

2.2.5 Multi-Dimensional Variational Calculus

The extreme problems depending on multi-dimensional functions are also very popular in many engineer and physical projects. Such problems are briefly introduced in the following section.

Theorem 2.7 Suppose D is a region and $(x,y)\in D,u(x,y)\in C^2(D)$, the necessary condition of minimising the functional

$$J(u(x,y))=\iint_D F(x,y,u,u_x,u_y)dxdy\,, \quad (2.47)$$

with the values of $u(x,y)$ given on the boundary ∂D of the region D, is that the extreme curves satisfy the Euler's equation in 2D case

$$F_u-\frac{\partial}{\partial x}F_{u_x}-\frac{\partial}{\partial x}F_{u_x}=0\,. \quad (2.48)$$

■

Corollary 2.3 (a) Suppose D is a region and $(x,y)\in D, u(x,y)\in C^{2n}(D), F\in C^{n+1}(D)$, the Euler's equation for the functional

$$J(u(x,y))=\iint_D F(x,y,u,u_x,u_y,u_{xx},u_{xy},u_{yy},\cdots,u_{\underbrace{xx\cdots x}_{n}},\cdots,u_{\underbrace{yy\cdots y}_{n}})dxdy, \tag{2.49}$$

is

$$\begin{aligned} &F_u-\frac{\partial}{\partial x}F_{u_x}-\frac{\partial}{\partial x}F_{u_x}+\frac{\partial^2}{\partial x^2}F_{u_{xx}}+\frac{\partial^2}{\partial x\partial y}F_{u_{xy}}+\frac{\partial^2}{\partial y^2}F_{u_{yy}}+\cdots+\\ &(-1)^n\left(\frac{\partial^n}{\partial x^n}F_{u_{\underbrace{xx\cdots x}_{n}}}+\frac{\partial^n}{\partial x^{n-1}\partial y}F_{u_{\underbrace{xx\cdots xy}_{n-1}}}+\cdots+\frac{\partial^n}{\partial y^n}F_{u_{\underbrace{yy\cdots y}_{n}}}\right)=0 \end{aligned}. \tag{2.50}$$

■

Corollary 2.3 (b) Suppose D is a region, $(x,y)\in D, u(x,y)\in C^2(D), v(x,y)\in C^2(D)$, the Euler's equation for the functional

$$J(u(x,y),v(x,y))=\iint_D f(x,y,u,v,u_x,v_x,u_y,v_y)dxdy, \tag{2.51}$$

is

$$\begin{cases} F_u-\frac{\partial}{\partial x}F_{u_x}-\frac{\partial}{\partial x}F_{u_x}=0\\ F_v-\frac{\partial}{\partial x}F_{v_x}-\frac{\partial}{\partial x}F_{v_x}=0 \end{cases}. \tag{2.52}$$

■

Corollary 2.3 (c) Suppose Ω is a *n-D* space and $(x_1,x_2,\cdots x_n)\in\Omega, u(x_1,x_2,\cdots x_n)\in C^{2n}(\Omega)$, the Euler's equation for the functional

$$J(u(x_1,x_2,\cdots x_n))=\int_\Omega F(x_1,x_2,\cdots,x_n,u,u_{x_1},u_{x_2},\cdots u_{x_n})dx_1dx_2\cdots dx_n, \tag{2.53}$$

is

$$F_u-\frac{\partial}{\partial x_1}F_{u_{x_1}}-\frac{\partial}{\partial x_2}F_{u_{x_2}}-\cdots-\frac{\partial}{\partial x_n}F_{u_{x_n}}=0. \tag{2.54}$$

■

2.3 Closure

In this chapter, theory of scale space and some related calculus of variations are discussed as well as some examples. The scale space theory is the foundation of multi-scale analysis. The relation between the scale space and PDE methods in image processing is addressed by Theorem 2.1 and further discussion is also given in later chapters. The anisotropic diffusion model discussed in later chapters is based on such theory. Calculus of variations, as a very important tool, is used in both image denoising and segmentation models as demonstrated in later chapters. Necessary condition for the existence of extrema is included to form the foundation work in later chapters. Theorem 2.7 and corollary 2.3(a) are used in most of chapters. For other cases, various researchers have dealt with them in other applications.

Chapter 3. OVERVIEW OF IMAGE PROCESSING

Image processing techniques have been developed over the last sixty to seventy years. There were applications in many satellite images and space images. Techniques were developed initially for analog images and later for digital images. Digital image processing is the technology of transferring an analog signal to a digital signal that can be used by computers in order to make the image more suitable for human observation and recognition. With the development of computer hardware digital image processing focuses on the applications of human vision, object detection, radar imaging, ultrasound imaging, computed tomography, magnetic resonance imaging, computer graphics, remote sensing and so on [65]-[67], by combining knowledge in many different areas such as optics, electronics, mathematics, computer science. In recent years, many new algorithms which make use of digital signal processing, information theory are introduced to make fast development in image processing. Some of these algorithms take the advantages of neural network, genetic algorithm, and artificial intelligence exploiting new approaches for image processing. Such efforts extend the range and potentiality of image processing in many industrial applications. Nowadays, branches of image processing include image transformation, image enhancement, image restoration, image segmentation, object recognition and image registration, image analysis, image reconstruction and image coding amongst others.

3.1 Image processing

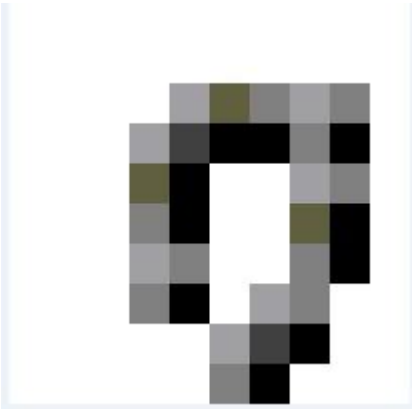
3.1.1 Image Function

An image function is a representation of an image in mathematics which can be a vector-valued function with regards to some arguments. In other words, it is the underlying

abstraction of an image. Take for an example, in computer vision, an image is modelled as a vector-valued function of integers for a colour image, or a function of integers for a grey image, depending on two spatial variables. In such case, mathematically, a two dimensional image u can be defined as

$$u:(x,y)\to V\subset\mathbb{R},\ (x,y)\in\Omega\subset\mathbb{R}^2, \tag{3.1}$$

If u is a colour image, V is a bounded vector space. Otherwise it is a set of real numbers, i.e. $V=\{g:g\in[0,255],g\in Z\}$ for 8-bit grey images, or $V=\{g:g\in\{0,1\}\}$ for morphological images. In this thesis, examples are mainly based on grey scale image as discussed. For convenience, in the following sections, an image is represented by $u(x,y)$. In the sense of discrete representation, an image is usually considered as a matrix in two dimensions as depicted in Figure 3.1.



(a) a zoom-in image

$$\begin{bmatrix}
255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255\\
255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255\\
255 & 255 & 255 & 255 & 247 & 91 & 164 & 247 & 164 & 255\\
255 & 255 & 255 & 247 & 82 & 0 & 0 & 164 & 0 & 255\\
255 & 255 & 255 & 91 & 0 & 255 & 255 & 247 & 164 & 255\\
255 & 255 & 255 & 164 & 0 & 255 & 255 & 91 & 0 & 255\\
255 & 255 & 255 & 247 & 164 & 255 & 255 & 164 & 0 & 255\\
255 & 255 & 255 & 164 & 0 & 255 & 247 & 164 & 255 & 255\\
255 & 255 & 255 & 255 & 255 & 247 & 82 & 0 & 255 & 255\\
255 & 255 & 255 & 255 & 255 & 164 & 0 & 255 & 255 & 255
\end{bmatrix}$$

(b) the corresponding grey values of (a)

Figure 3.1: Image discrete representation.

3.1.2 Image Processing Model

As mentioned at the beginning of this chapter, image processing has different branches. However, when doing image processing, all of these processes can be abstractly formulated by an input-output system [68]:

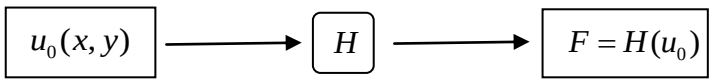


Figure 3.2: Image processing input-output system.

Here the input u_0 means an original image, which could have been degraded during poor acquisition of the image, storage, or communication that needs to be processed. Sometimes

for complicated cases, there is more than one single image or other media, such as video clips. H is an image processor, for example, that may be capable of performing denoising, deblurring or segmentation, which can be any linear or nonlinear operator, mathematically. The design and computing of H are the most important tasks during the processing. The main challenge is that most image processing problems are ill-posed inverse problems, which leads to non-unique solution and instability of the method. F is a desirable target which depends on specific applications or tasks. It can be generally any set of features which is visually meaningful, such as a restored image for image restoration, certain feature of an image for feature extraction, or certain pattern for image analysis.

One typical example of the image processor is the image denoising model. This model builds a degenerated model according to some prior knowledge of a degraded image and then uses some tools, such as filtering, to restore the degraded image in order to obtain a favourable result. In this case, u_0 means a given noisy image and u is a clean, but usually unknown image, i.e.

$$u_0 = Hu + n, \tag{3.2}$$

where H is a linear operator representing the blur (usually a convolution) and n is an additive noise. Then the task is to design an image processor which can restore u from u_0.

3.2 Image Noise

Image noise involves random errors of brightness or colour information in images which are introduced into images due to many different processes, including the process of acquiring and transferring images, the process of digitalisation, etc. Such noise is usually unwanted, unpredictable and its properties may be studied by means of probability and statistical methods. The presence of noise leads to undesirable appearance of an image and can affect and reduce the visibility of certain features in the image. Therefore, image denoising becomes an essential part in image pre-processing. Actually, it is one of the oldest techniques in image processing and still receiving considerable attention from both engineers and mathematicians. An example of noise is shown in Figure 3.3.

Figure 3.3: A noisy image[1].

Generally, noise in image processing can be categorized into two types. They are additive noise, such as Gaussian noise as shown in (3.2) and multiplicative noise, such as speckle noise as shown in (3.3).

$$u_0 = u \cdot n. \tag{3.3}$$

More specifically, for the convenience of studying the properties of different kinds of noise, there are some image noise models which are widely accepted by many researchers. For example, Gaussian noise, salt and pepper noise, Rayleigh noise, Erlang (Gamma) noise, exponential noise, uniform noise, heavy tailed noise, photon counting noise, photographic grain noise. Every noise model has its own probability density function (PDF). For instance, the PDF of Gaussian noise is shown in

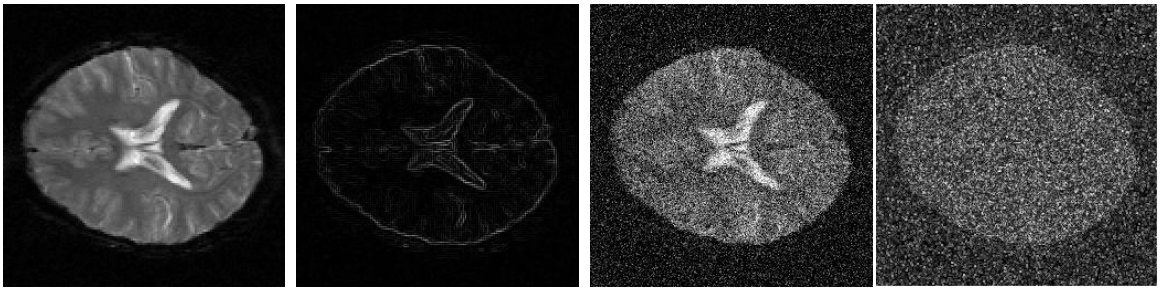
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad x \in (-\infty, +\infty), \tag{3.4}$$

Here, x is the grey value of noise, μ and σ^2 denote the expectation and variance of the noise respectively. In this model, σ^2 is a significant parameter showing the degree that an image is affected by noise.

Although in some occasions, noise can provide special effects in some area, such as the old fashioned, grainy look in art work, it usually degrades the quality of an image and makes the image undesirable. Furthermore, the existence of noise may affect the results of other steps in

[1] Except for some self-designed images, described as accordingly, images used for experiments in this thesis were taken from http://www.imageprocessingplace.com

image processing enormously. For example, when using canny operator for edge detection, the existence of noise may make the edge detection fail in the worst scenario (Figure 3.4). Therefore, efficient noise suppression enables the success of the subsequent image processing and its analysis.



(a) original image (b) edge detection of (a) (c) (a) with noise (d) edge detection of (c)

Figure 3.4: The influence of noise in the procedure of edge detection.

Nowadays, image denoising can be done by many methods, such as filtering, wavelets, PDE methods, variational methods and so on. In this thesis, PDE methods are used to remove Gaussian noise from an image.

3.3 Methodologies in Image Processing

In this section, a brief introduction is given of several current approaches to image processing, including filtering, mathematical morphology, wavelet analysis, stochastic modelling, PDE-based methods and variational methods. Further details about these different approaches can be found from [69]-[78] and citations therein.

3.2.1 Filtering

As one of the earliest techniques in image processing, filtering methods are useful and simple and still suitable for many applications [69]. A filter is defined by a kernel, which is a small two dimensional array (usually it is known as a window) applied to every pixel and its neighbours in an image. For example, the mean filter can be implemented using the weighted sum of the pixels in successive windows, i.e.

$$h(x,y)=\frac{1}{M}\sum_{(k,l)\in N}u(k,l), \tag{3.5}$$

where M is the total number of pixels in the neighbourhood N. The process of employing linear filters to an image is known as convolution and can be applied to both spatial and frequency domain, which is feasible for many applications. Traditionally, filtering methods can be classified into two kinds of filtering, linear filtering (e.g. mean filter), and nonlinear filtering, (e.g. median filter). Each filter has its own specific function. For examples, median filter is mostly used for noise reduction and an unsharp filter for enhancement. Moreover, the filtering has a close relation with other methods in image processing, such as isotropic diffusion model in PDE-based methods can be derived from filtering methods, which will be introduced in the coming chapters.

3.2.2 Mathematical Morphology

The concept of mathematical morphology is based on the fact that an image consists of pixels that collect into groups having a two-dimensional shape. It aims to analyse the shapes and forms and extracts image components which are useful in the representation and description of an object [71]. This method was initially developed for binary images, and then due to its specific features, was later extended to grey-scale images. It concerns nonlinear image transformations such as erosion, dilation, opening, closing, and morphological skeleton by taking advantage of the size, shape, convexity, connectivity and geodesic distance of an image. In industrial applications, this method is widely used in image enhancement, image segmentation, object recognition, and shape analysis amongst others.

3.2.3 Wavelet Analysis

Wavelet, as a multi-scale analysis tool, may be used to separate data into several frequency components and then investigates each component according to its corresponding scale [74]. The procedure of wavelet analysis includes temporal analysis and frequency analysis. The former is performed with a shrinking, high-frequency version of the prototype wavelet, i.e. mother wavelet, while the latter is conducted with a dilated, low-frequency version of the same wavelet. This method deals with data at different scales or resolutions. It can not only handle details, but also contours of an object. Compared with Fourier transformation, wavelet method performs much better in analysing physical situations where the signal contains discontinuities and sharp spikes. This method was developed independently in the areas of quantum physics, mathematics, seismic geology and electrical engineering, which leads to

many applications such as image processing, radar, turbulence and earthquake prediction and so on.

3.2.4 Stochastic Modelling

Stochastic modelling is a statistical process to evaluate probability distributions of potential outputs by allowing for random variation in one or more inputs over time [75]. Based on Markov random field theory, it allows uncertainties to be taken into account to provide a robust representation of the current problem. In many fields of image processing, there is usually a large degree of uncertainty factors such as appearance of noise within the acquired image, the location of interest in an object etc. Hence, for images with notable stochastic nature, statistical approaches are quite more suitable than deterministic methods and their relevant tools for modelling image processing. Not only is it used in medical image processing, but also widely spreads to many other applications including image denoising, image segmentation, image registration, and feature extraction. With this method, images can be processed in a robust and efficient manner making the procedure more desirable to the specific application.

3.2.5 PDE-based Methods

The concept of using partial differential equations in image processing dates back to the 1960s. It is now being used in much image processing work ranging from image denoising, segmentation to optical flow and even 3D modelling. PDE-based methods are formulated as diffusion processes initially which can take full advantage of local structures. Overtime, various PDE-based methods have been proposed via different ways. For example, models directly based on mathematical properties of a PDE (e.g., anisotropic diffusion [9]) or its application in other area [79], energy functional defined for different purposes leading to optimal solutions written PDEs, and axiomatic approaches leading to PDE from scale space theory. All of these ways need good skills in choosing proper functions or operators (gradient, Laplacian, etc.) to make the derived PDE meet the final target. Nowadays, these methods appear in a large variety of areas, ranging from anisotropic diffusion, active contour models, shape-from-shading to optical flow, etc. All of these cover image denoising, image inpainting, image segmentation, image registration, 3D model reconstruction, which give a very bright future for PDE-based methods.

3.2.6 Variational Methods

During the last two decades, a new approach in image processing techniques has become more and more popular, which formulates the problem by using variational methods incorporating suitable physical concepts into the formulation. These methods employ a cost functional or energy functional which presents certain variable in an intuitive and natural way. The functional is optimised with respect to certain constraints leading to a partial differential equation, which can then be solved by a number of numerical schemes. On one hand, since the energy functional is feasible, it provides various possibilities to obtain a desired result. On the other hand, it is different from traditional image processing techniques due to its being formulated by optimisation problems. Therefore, an optimal solver is usually available for original problems. Combining with PDE technique, these methods have proven to be very powerful tools to solve many image processing tasks, especially in image denoising, image segmentation, image registration and implicit representation by level sets.

Those approaches presented above are the most common methods in recent years. Although different methods may belong to different subjects, there are some intrinsic interconnections between each other [68]. Every approach has its own advantages in dealing with image processing problems. In this thesis, PDE-based methods and variational methods are the main approaches employed. In the next section， PDE-based methods are discussed in more details because variational methods eventually also lead to PDEs.

3.3 PDE-based Methods and Image Processing

As mentioned in section 3.2.5, a partial differential equation for image processing may be induced by different ways, even from traditional methods such as filtering. In this section, several ways to obtain such PDE are investigated and the advantages of using PDE in image processing are discussed.

3.3.1 From Scale Space to PDE

In Example 2.1, an example of the relationship between scale space and PDE is briefly described. In this section, a detailed examination on such derivation is given of the linear scale space (Gaussian scale space) using constraints and invariant properties. The example is restated as below:

Given a scale space $\{T_t\}$ defined by $T_t u = u(t)$ $T_t u = u(t)$, such that the operator T_t is causal, linear, translation and Euclidean invariant, then $u(t)$ is the solution of the heat equation

$$\begin{cases} \dfrac{\partial u(t)}{\partial t} = \nabla^2 u(t) \quad (\mathbf{x}, t) \in \Omega \times (0, T] \\ u(0) = u_0 \end{cases}, \tag{3.6}$$

where u_0 is the initial value.

Proof:

Suppose F is independent of t. Since $\{T_t\}$ is translation invariant, thus using Theorem 2.3 leads to $F(A, P, x, c) = F(A, P, c)$. Furthermore, $\{T_t\}$ is linear, then $F(A, P, c) = F(A, P)$. Due to its regularity, $F(A, P) = \lim_{t \to 0} \dfrac{T_t u - u}{t}$. Because $\{T_t\}$ is linear, F is linear with respect to u. Therefore, $\forall r, s \in \mathbb{R}$, vectors p, p' and symmetric matrices A, A' satisfy the relation

$$F(rA + sA', rp + sp') = rF(A, p) + sF(A', p'), \tag{3.7}$$

and

$$F(A, p) = F(A, \mathbf{0}) + F(\mathbf{0}, p). \tag{3.8}$$

Rewrite (3.8) as

$$F(A, p) = F_1(A) + F_2(p). \tag{3.9}$$

According to Theorem 2.4, it can be obtained that

$$F(RAR^T, Rp) = F(A, p), \tag{3.10}$$

where R is an arbitrary isometric transformation. Let $A = \mathbf{0}$, from (3.9) and (3.10), it can be deduced that for arbitrary p and symmetric matrix A

$$F_1(RAR^T) = F_1(A), \tag{3.11}$$

and

$$F_2(Rp) = F_2(p) \tag{3.12}$$

It is known that symmetric matrices can be diagonalised by an orthogonal basis and every orthogonal basis can be changed by an isometric transformation. Therefore, the function F_1 is only dependent on the eigenvalues of *A*, i.e., $\lambda_1, \lambda_2, \cdots, \lambda_n$, that is,

$$F_1(A) = F_1(\lambda_1, \lambda_2, \cdots, \lambda_n). \tag{3.13}$$

Thus, F_1 is a function which solely depends on the eigenvalues of *A*. According to [82], F_1 can be expressed as

$$F_1(A) = c \times trace(A), \tag{3.14}$$

where *c* is a constant. On the other hand, F_2 is linear and from (3.12), it can be shown that for arbitrary vector *p*,

$$F_2(p) = 0. \tag{3.15}$$

Therefore, from (3.9), (3.14), and (3.15)

$$F(A, p) = c \times trace(A). \tag{3.16}$$

Without loss of generality, *c* can be taken as 1. Furthermore, based on the definition of regularity, $A = D^2u$, $p = Du$, then $trace(A) = \nabla^2 u$. Consequently, it can be obtained that

$$F(D^2u, Du) = \nabla^2 u. \tag{3.17}$$

Thus, the heat equation is derived.

If *F* depends on *t*, identically, it can be shown that

$$F(D^2u, Du, t) = c(t)\nabla^2 u \tag{3.18}$$

In such case, a new scale parameter $\tau(t)$ can be redefined as $\frac{d\tau}{dt} = c(t)$ such that a new heat equation is obtained as

$$\frac{\partial u}{\partial \tau} = \nabla^2 u. \tag{3.19}$$

■

In summary, the conclusion of the example above can be shown by Figure 3.5.

$$T_t \begin{cases} \text{Linear} \\ \\ \text{Casual} \end{cases} + \begin{cases} \text{Euclidean invariant} \\ \\ \text{Translation invariant} \end{cases} \Rightarrow \quad \frac{\partial u}{\partial t} = \nabla^2 u$$

Figure 3.5: Linear scale space and PDE

3.3.2 From Gaussian Convolution to PDE

Gaussian function was initially used in image processing as a filtering since 1980 by Marr and Hildreth [83]. The main idea of Gaussian filtering is to apply a convolution between an initial image $u_0(x)$ and a Gaussian kernel $G(x,t)$. By changing the parameter *t*, one can get a family of images in different scales. Based on such idea, Witkin [9] and Koenderink [10] developed the Gaussian filter to Gaussian scale space and it was proved that Gaussian scale space is the unique linear scale space which is linear, casual, translation and Euclidean invariant [84]. In this section, the relationship between Gaussian convolution and PDE is examined.

Consider the one dimensional Gaussian function

$$G(x,t) = \frac{1}{\sqrt{4\pi t}} \exp\left(-\frac{x^2}{4t}\right), \tag{3.20}$$

and an initial image $u_0(x)$, the convolution of $u_0(x)$ and $G(x,t)$ leads to the resulting image

$$u(x,t) = \left(G(x,t) * u_0\right)(x) = \int_\Omega G(x-y,t)u_0(y)dy, \tag{3.21}$$

which is also the solution of the heat equation (3.6).

Applying Fourier transform ($\hat{f}(\omega) = \int_\Omega f(x)e^{-i\omega x}dx$) on (3.6) leads to

$$\begin{cases} \frac{\partial \hat{u}}{\partial t}(\omega,t) = |\omega|^2 \hat{u}(\omega,t) \\ \hat{u}(\omega,0) = \hat{u}_0(\omega) \end{cases}. \tag{3.22}$$

Then

$$\hat{u}(\omega,t) = c \cdot e^{|\omega|^2 t}. \tag{3.23}$$

Let $t = 0$, then $c = \hat{u}_0(\omega)$. Thus, (3.23) becomes

$$\hat{u}(\omega,t) = \hat{u}_0(\omega) \cdot e^{|\omega|^2 t}. \tag{3.24}$$

Secondly, applying the inverse Fourier transform ($f(x) = \frac{1}{2\pi}\int_{\Omega} \hat{f}(\omega) e^{i\omega x} d\omega$) on (3.24), one can obtain

$$u(x,t) = G(x,t) * u_0 \ (x) \tag{3.25}$$

■

As a consequence, using a Gaussian filter in image processing plays the same effect as solving the heat equation. Therefore, one can use the heat equation and other PDEs to deal with problems in image processing.

3.3.3 From filtering to PDE

In image processing, a filter is usually defined as a two dimensional array which can be presented by a template or mask below,

$$T = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}. \tag{3.26}$$

The filtering method can be expressed by using a discrete convolution

$$u(m,n) = T(m,n) * u_0(m,n) = \sum_{i=-r}^{r} \sum_{j=-s}^{s} T(i,j) u_0(m-i, n-j), \tag{3.27}$$

where *r* and *s* depend on the size of template *T* and *u* is the generated image from the original image u_0.

Consider the neighbourhood averaging filter with the template

$$T = \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}. \tag{3.28}$$

After convolution, the generated image is

$$u(i, j) = \frac{1}{4}[u_0(i-1, j) + u_0(i+1, j) + u_0(i, j-1) + u_0(i, j+1)]. \tag{3.29}$$

If the filtering process is repeated until a certain criterion is satisfied, then u_0 in (3.29) can be replaced by *u*, and then this process can be seen as an iterative process. Therefore, (3.29) can be rewritten as

$$u(i, j) = \frac{1}{4}[u(i-1, j) + u(i+1, j) + u(i, j-1) + u(i, j+1)]. \tag{3.30}$$

Rearrange (3.30) gives

$$u(i-1, j) + u(i+1, j) + u(i, j-1) + u(i, j+1) - 4u(i, j) = 0. \tag{3.31}$$

(3.31) is a finite difference replacement of the Laplace equation:

$$\nabla^2 u = 0. \tag{3.32}$$

Generally, (3.32) can be solved by a pseudo time-dependent PDE with a given initial condition, which is the same as that in (3.6).

■

In this way, the PDE may be regarded as being induced from the filtering methods. However, not all filters have their equivalent continuous forms. For those filters that have a continuous form of a PDE, it is possible to introduce further new features into the PDE itself to other modified or enhanced techniques in order to handle other image processing. Such features may not be introduced through the use of filters.

3.3.4 Advantages of PDE

As mentioned in section 3.2, there are many approaches to do image processing. Comparing with other methods in image processing, generally speaking, PDE-based methods have several advantages [80].

- PDE-based methods can take full advantage of local structure (i.e., gradient, Laplacian) as well as geometrical information (i.e., curvature), etc. The information contains the main features of an image. By using such information, PDE-based methods can serve efficiently for image processing.
- PDE-based methods lead to modelling images in a continuous domain. When an image is represented as a continuous signal, PDE-based methods can be considered as an iteration of local filters which allows to unify and classify a number of existing iterative filters as well as to develop new ones. Furthermore, in a continuous domain, existing theories for PDEs are well established, which lays a good theoretical foundation for image processing.
- PDE-based methods are efficient, accurate and stable. The extensive literature available on numerical analysis about PDEs provides diverse algorithms for implementing PDE-based methods and useful theoretical results such as existence and uniqueness of the solution.
- PDE-based methods can separate the theoretical analysis and implementation process. When using PDE-based methods in image processing, the theory and feature analysis are conducted in the continuous domain while the computation is implemented in a discrete domain (pixels). This would greatly facilitate the use of existing PDE theory in understanding relate image processing techniques.

These advantages contribute to the fact that much research has been carried out in the use of PDE methods in image processing leading to much understanding in image restoration, segmentation, inpainting and acquired very satisfying results([46], [57], [68] and citations therein).

Chapter 4. **PDE-BASED IMAGE DENOISING**

Noise in an image appears as an unintended component and significantly degrades the image quality. Denoising is an essential step for image pre-processing. A number of different techniques have been examined over the last decades. These include filtering, wavelets, PDE-based approaches and variational methods. The use of PDE in image processing has grown significantly over the past few years due to the many advantages mentioned in the previous chapter.

This chapter reviews some classical models using PDE in image smoothing and restoration, including isotropic diffusion model, anisotropic diffusion model, and the fourth-order PDE model, and sketches the underlying ideas of them. It aims to provide a brief historical picture of the main PDE-based methods in image processing studied and improved in this thesis.

4.1 PDE-Based Methods

The basic idea of using PDE in image processing is to evolve an image, a curve, or a surface using a PDE, and obtain the anticipated results. Mathematically, the process of using PDE in image restoration can be described as below.

Given an image $u_0 : \mathbb{R}^2 \to \mathbb{R}$ and its grey values are $u_0(x, y)$, the mathematical model for image restoration by using PDE can be written as

$$\frac{\partial u}{\partial t} = F(u(x, y, t)), \tag{4.1}$$

where $u(x, y, t) : \mathbb{R}^2 \times [0, T) \to \mathbb{R}$ is the image at time *t*. $F : \mathbb{R} \to \mathbb{R}$ is a mapping that applies to an image, which usually depends on the image itself and the first and second derivatives of

the image, i.e. u, ∇u, $\nabla^2 u$, etc. The resulting image obtained by this PDE is generally an approximation to the original clean image.

4.2 Isotropic Diffusion (Linear Diffusion)

As the simplest PDE method for image restoration, the isotropic diffusion or linear diffusion is worth being investigated to obtain the underlying ideas of PDE methods in image processing. In Chapter 3, the relationship between Gaussian convolution and isotropic diffusion is discussed and in this section the physical background, models and performance of isotropic diffusion are studied.

4.2.1 Physical Background

In physics, the physical quantity which steers the heat transfer due to the asymmetry of temperature distribution in a conductive medium is known as the heat conductivity or heat diffusivity. Given a temperature distribution u, the diffusive flux $\mathbf{j}$, which measures the amount of temperature going from high temperature to low temperature per unit time due to diffusion, is governed by Fick's law

$$\mathbf{j} = -c\nabla u . \tag{4.2}$$

Here $\mathbf{j}$ is the diffusive flux and c is known as the diffusion coefficient, which relates the diffusive flux and the gradient of the temperature. This coefficient can be a constant, a symmetric matrix, or a scalar function with respect to certain property of the material. It expresses the fact that the gradient ∇u leads to a flux $\mathbf{j}$ which compensates this gradient. The diffusion that only transfers heat without loss or adding new quantity of heat into the material is governed by the continuity equation,

$$\frac{\partial u}{\partial t} = -\nabla \cdot \mathbf{j}, \tag{4.3}$$

where t denotes the time. Putting (4.2) into (4.3), one obtains

$$\frac{\partial u}{\partial t} = \nabla \cdot (c\nabla u) . \tag{4.4}$$

Supplemented with suitable boundary conditions, this equation reveals the diffusion of heat energy in the temperature field and settles the temperature of the conductive medium into equilibrium.

Apart from heat transfer, (4.4) has applications in many other transport phenomena, e.g. concentration diffusion and image smoothing. The latter is described as below.

4.2.2 Modelling and Scale Property

In the light of idea described above, Koenderink [10] borrowed the heat conduction equation for image denoising by taking u as the grey-value function and by setting c as 1 in an image support $\Omega \subset \mathbb{R}^2$ in order to diffuse certain random noise in the mage. For this purpose, (4.4) can be rewritten as

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla^2 u, & (x, y) \in \Omega \\ \dfrac{\partial u}{\partial n} = 0, & (x, y) \in \partial\Omega, \\ u(x, y, 0) = u_0(x, y), & t = 0 \end{cases} \tag{4.5}$$

where n is the inward normal vector to the boundary $\partial\Omega$. Here t is also known as the scale parameter in the context of image processing. As discussed in chapter 3, it has been shown that the solution of this equation can also be obtained by the convolution between a Gaussian kernel and $u_0(x, y)$, that is

$$u(x, y, t) = G_t(x, y) * u_0(x, y), \tag{4.6}$$

where $G_t(x, y) = at^{-1} \exp[-(x^2 + y^2)/4t]$ is the Gaussian kernel, and a is a constant. It is assumed here that the diffusion is the same in all directions for all scales. Thus this diffusion model is well known as isotropic diffusion model (linear diffusion) or Gaussian filter in image processing. It considers an image as a panel with different heat quantity, and the temperature of the panel becomes equilibrium as time goes on.

It is well known that images contain structures at a large range of scales as discussed in chapter 2. In most cases, it is not apparent that which scale is desirable to represent a best image. However, multi-scale images still provide a hierarchy of image structures which may be useful for image interpretation. As a simple filter, Gaussian filter builds a scale space,

known as Gaussian scale space, by increasing variance, or in other words, by the linear diffusion process shown in (4.5). Such scale space has two properties [10]:

(a) Causality: The scale space operator brings no spurious details when increasing the scale parameter;

(b) Homogeneity and Isotropy: The scale space operator considers all spatial points and scale levels equally.

4.2.3 Numerical Implementation

In order to implement the isotropic diffusion numerically as well as other models shown in this thesis, a numerical framework is built here and also for the chapters coming after. Consider an image function defined on a discrete lattice in the form of pixel as depicted in Figure 4.1.

$u_{i-1,j-1}$	$u_{i-1,j}$	$u_{i-1,j+1}$
$u_{i,j-1}$	$u_{i,j}$	$u_{i,j+1}$
$u_{i+1,j-1}$	$u_{i+1,j}$	$u_{i+1,j+1}$

Figure 4.1: Discretisation of an image.

Given that the size of the image is $Ih \times Jh$, where h denotes the size of the lattice. Let Δt be the time step. The discretisation of the image support Ω leads to a network of lattice with coordinates located at

$$(x_i, y_j) = (ih, jh),\ i = 1, 2, \cdots, I,\ j = 1, 2, \cdots, J\ .$$

And the pixel intensity at (x_i, y_j) is denoted as $u(x_i, y_j)$, or simply $u_{i,j}$. Using the notation above, the discretisation of the first and second order derivatives of an image can be written as

$$\frac{\partial u}{\partial x}\Big|_{(i,j)} = \frac{u_{i+1,j} - u_{i,j}}{h}, \tag{4.7}$$

and

$$\frac{\partial^2 u}{\partial x^2}\Big|_{(i,j)} = \frac{u_{i,j+1} + u_{i,j-1} - 2u_{i,j}}{h^2}. \tag{4.8}$$

Therefore, the norm of gradient and Laplacian operator can be discretised as

$$|\nabla u_{i,j}| = \left[\left(\frac{u_{i+1,j} - u_{i,j}}{h} \right)^2 + \left(\frac{u_{i,j+1} - u_{i,j}}{h} \right)^2 \right]^{\frac{1}{2}}, \tag{4.9}$$

and

$$\nabla^2 u_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2}. \tag{4.10}$$

With these discretisations, (4.5) can be discretised as

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \frac{u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}}{h^2}. \tag{4.11}$$

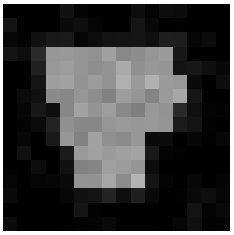
Rearrange (4.11), an iterative process can be obtained by using the explicit form

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t \frac{u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}}{h^2}. \tag{4.12}$$

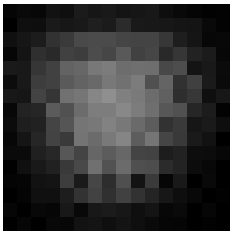
The default size of lattice is chosen as 1, that is, $h = 1$, in the experiments described in the remaining of this thesis.

4.2.4 Conclusion

The isotropic diffusion opens up a number of new approaches, that is, PDE-based methods, for image processing. Due to its linearity, this model can be easily handled and implemented. However, this model not only smoothes the noise in an image, but also blurs other information, such as edges. The reason is that the model does not take account of prior knowledge of the image. Technically, this diffusion process is a low-pass filtering. Such filtering suppresses the high-frequency parts, such as noise and edges, while preserving the low-frequency parts of an image. As a consequence, this makes edges blurring (the extent of the blur depends on the choice of the variance in the Gaussian kernel) and difficult to be located. Such phenomenon is known as Gaussian blurring in image processing as depicted in Figure 4.2.



(a) the original image



(b) the denoised image , 5 iterations

Figure 4.2: Image denoising by using isotropic diffusion model.

4.3 Anisotropic Diffusion

4.3.1 Motivation and Modelling

The diffusion caused by (4.5) takes place irrespective of the pixel point for all scales and does not distinguish noise and certain image features, which leads to Gaussian blurring as discussed before. To overcome this deficiency, Perona and Malik proposed three rules of describing and handling multi-scale properties of image smoothing [12]. These are

1. Causality,
2. Immediate Localization, and
3. Piecewise Smoothing.

The causality implies that when t increases, there is no spurious detail generated and the features of the image diminish with the enhancing smoothing effect. The immediate localization states that when smoothing images, the spatial location of image boundaries would not change and the boundary features can be located precisely for all scales. The piecewise smoothing takes boundaries into consideration. Smoothing within a region should occur preferably over smoothing across the boundaries. Perona and Malik used a spatial varying coefficient $c(s)$ instead of a constant as in (4.4). The aim is to reduce the diffusion coefficient while crossing potential edges. This model is known as the P-M model for short and is an anisotropic diffusion given by

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla \cdot (c(s)\nabla u), & t > 0,\ (x,y) \in \Omega \\ \dfrac{\partial u}{\partial n} = 0, & (x,y) \in \partial\Omega \\ u(x,y,0) = u_0(x,y), & t = 0 \end{cases}, \tag{4.13}$$

where $s = s(x, y, t)$ contains certain information of an image and $c(\cdot)$ is the diffusion coefficient. The key point of this model is the choice of the coefficient to attain faster smoothing in noisy area than that across edges. If $c(s) = 1$, it reduces to the isotropic diffusion equation as used by Koenderink. If $c(\cdot)$ depends on the image, the diffusion coefficient is changed to a nonlinear diffusion which makes the method an adaptive smoothing method.

4.3.2 Choices of the Coefficient

In order to satisfy the two rules described above, i.e., the immediate localization and piecewise smoothing, the diffusion coefficient $c(s):[0,+\infty] \to [0,1]$ has to be a non-negative monotonically decreasing function with $c(0) = 1$ and $\lim_{s \to \infty} c(s) = 0$ as depicted in Figure 4.3. With such a coefficient, the diffusion process can smooth an image selectively. As a consequence, this accelerates the diffusion in almost flat regions of the image and reduces the diffusion speed across edges.

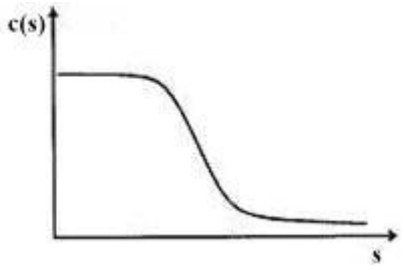


Figure 4.3: The qualitative shape of the nonlinear coefficient $c(\cdot)$.

The diffusion coefficients suggested by Perona and Malik are given below.

$$c(s) = c(\| \nabla u \|) = \frac{1}{1 + (\frac{\| \nabla u \|}{K})^2}, \tag{4.14}$$

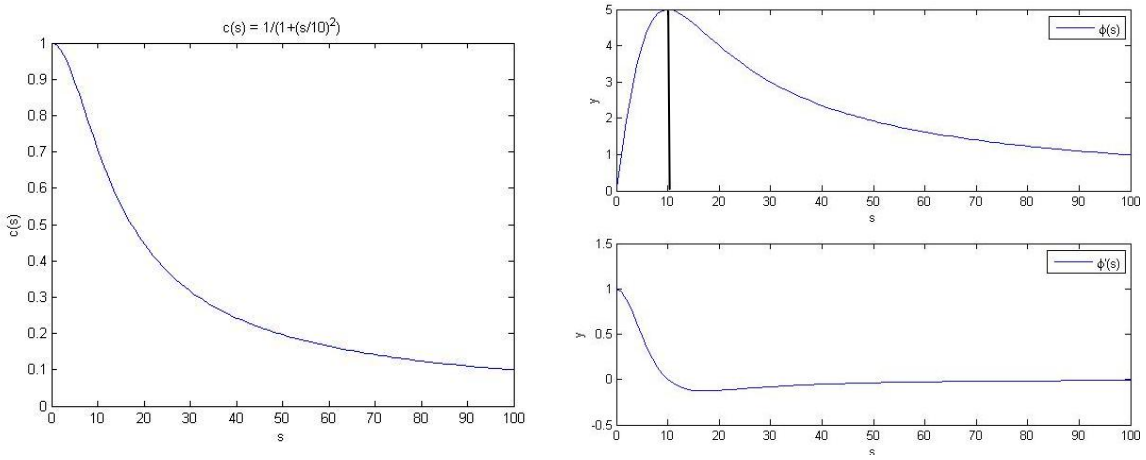
and

$$c(s) = c(\| \nabla u \|) = e^{-(\frac{\| \nabla u \|}{K})^2}, \tag{4.15}$$

where K is a threshold to control the diffusion process. In both coefficients, as s increases, $c(s)$ decreases and tends to zero around image edges and thus prevents diffusion at the edges. In this thesis, (4.14) is adopted to study the P-M model. The ϕ-variable defined below may be used to explain the role of the threshold K.

$$\phi(s) = sc(s) \tag{4.16}$$

Figure 4.4(a) shows the diffusion coefficient defined in (4.14) and the corresponding ϕ-variable defined in (4.16) is shown in Figure 4.4 (b) for the case when $K = 10$. It can be easily seen that (4.14) is a non-negative monotonically decreasing function.



（a） $c(s) = \dfrac{1}{1+(\dfrac{s}{K})^2}$, K = 10,　　（b）top: $\phi(s)$, bottom: $\phi'(s)$ K = 10,

Figure 4.4: Shapes of $c(s)$ and $\phi(s)$.

From Figure 4.4, when $\|\nabla u\| < K$, the ϕ-variable increases as the norm of the gradient increases, whereas when $\|\nabla u\| > K$, the ϕ-variable decreases as the norm of the gradient increases. Two cases of extreme values of $\|\nabla u\|$ exist. First, when $\|\nabla u\| << K$, the smoothing is weak by virtue of small ϕ-variable in the homogeneous regions of the image. Second, when $\|\nabla u\| >> K$ which represents edges, the ϕ-variable is also small in order to preserve edges. Values of $\|\nabla u\|$ other than the two cases given above are considered as the noisy area. Thus the larger the value of the ϕ-variable the more smoothed is the area.

4.3.3 Behaviours of P-M model

The P-M model makes use of the local intensity information of the image involving simple operations. It comes up with a solution to keep the noise reduction while sharpening the edges. Therefore, this model is also suitable for edge detection. In order to understand the behaviours of this model, the one dimensional case is considered here. In such case, (4.13) becomes

$$u_t = \frac{\partial}{\partial x}\left[c\left(|u_x|\right)u_x\right], \tag{4.17}$$

which can be simplified as

$$u_t = \left[c'\left(|u_x|\right)|u_x| + c\left(|u_x|\right)\right]u_{xx}. \tag{4.18}$$

From Eqn (4.16), it can be seen that $\phi'(s) = sc'(s) + c(s)$, thus (4.18) can be further stated as

$$u_t = \phi'(|u_x|)u_{xx}. \tag{4.19}$$

The discrete form of (4.19) at point *p* could be stated as

$$u_p^{n+1} = u_p^n + \Delta t\phi'\left(\left|u_{p+1}^n - u_p^n\right|\right)(u_{p+1}^n + u_{p-1}^n - 2u_p^n). \tag{4.20}$$

If the coefficient $c(s)$ is chosen as (4.14), then

$$\phi'(s) = \frac{1-\left(\frac{s}{K}\right)^2}{\left[1+\left(\frac{s}{K}\right)^2\right]^2}. \tag{4.21}$$

Thus,

$$\begin{cases} \phi' \geq 0, 0 \leq s \leq K \\ \phi' < 0, s > K \end{cases}. \tag{4.22}$$

From Figure 4.4(b), the same result can be observed. Therefore, for $|u_x| < K$, that is, $\phi'(|u_x|) > 0$, if the grey value of u_p is smaller than those of its neighbouring points which makes the second term in (4.20) positive, the diffusion increases the grey value of u_p. Conversely, if the grey value of u_p is greater than those of its neighbouring points, the grey value of u_p is decreased by iterations. Such process is known as a forward diffusion which can smooth noise. For $|u_x| > K$, the same analysis can be employed and such process is named as a backward diffusion which can enhance edges.

However, although it can remove noise effectively and meanwhile preserve the edges, it generates the "block effect". The "block" is defined as regions of which the grey values are almost the same in each region. In one dimension case, this phenomenon is known as "staircasing". The reason of "block effect" is that when diffusing in homogeneous regions, the diffusion is strong and thus the noise is diminished. Nevertheless, in the areas of which the gradient is large, the diffusion is slow and thus the difference of grey values between these areas and homogeneous areas is relatively large. In such case, the grey values in a homogeneous region tend to be a constant and edges around this region are sharpened as the iteration increases. Eventually, the image appears as several piecewise smoothing regions, that is, the "block effect". Figure 4.5 (b) shows the "block effect" in a matrix after applying the P-M algorithm to the matrix in Figure 4.5 (a) (this matrix is created on purpose). According to the research by Yu and Kaveh [88], if there is no backward diffusion, the P-M model will evolve toward a level image. The "block effect" will emerge in the early stage of the diffusion and then all the blocks will merge to a level image ultimately. If there is backward diffusion, the "block effect" will appear from the beginning to the end. However, since the norm of gradient in an image is not infinite, an image will appear as a level image even with the backward diffusion, which is shown in Figure 4.5 (c). Meanwhile, it can also be observed that the inner area shrinks with the diffusion going on. This means fake edges may come out after the diffusion.

0	12	0	14	1	25	20	11	16	13	25	23	25	30	14	0
0	12	11	0	14	0	25	11	0	14	11	19	30	14	15	15
10	11	25	25	25	25	25	25	25	25	25	25	15	15	15	17
11	25	123	144	124	135	183	145	123	158	146	25	25	16	15	14
0	25	134	114	142	154	151	163	153	141	165	154	25	25	5	9
1	25	25	136	163	174	164	145	154	129	154	157	162	25	9	10
16	15	25	156	175	135	174	143	135	138	168	178	120	25	9	11
14	13	25	25	125	145	134	134	174	149	159	119	25	25	15	25
18	16	8	25	126	144	154	145	164	142	163	25	25	15	21	24
15	14	7	25	156	153	136	185	132	145	25	25	15	14	15	11
13	12	13	18	25	147	124	147	164	25	25	15	12	22	14	12
9	17	17	15	25	161	185	136	25	25	14	7	6	22	10	13
4	0	15	11	25	135	135	25	25	20	15	4	13	14	25	19
12	28	13	2	5	25	25	25	11	11	6	11	0	10	12	11
11	17	15	14	14	0	25	21	12	15	7	12	15	9	19	5
0	19	10	0	15	11	0	4	14	11	14	10	11	12	7	8

(a) a 16×16 matrix

0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	1	1	2	2	2	2	2	2	2	2	2	1	1	1	0
1	1	132	133	134	134	135	135	135	135	134	2	2	1	1	0
1	1	133	134	134	135	135	136	136	135	135	134	2	2	1	1
1	2	3	134	135	135	136	136	136	136	135	135	134	2	1	1
1	2	3	134	135	135	136	136	136	136	135	135	134	2	1	1
1	2	2	3	135	135	136	136	136	136	135	135	3	2	1	1
1	2	2	3	135	135	135	136	136	135	135	4	3	2	1	1
1	1	2	3	134	135	135	135	135	135	4	3	3	2	1	1
1	1	2	3	4	134	134	134	134	4	4	3	2	2	1	1
1	1	2	2	3	133	133	134	4	3	3	3	2	2	1	1
0	1	1	2	2	132	132	3	3	3	2	2	2	1	1	0
0	1	1	1	2	2	2	2	2	2	2	2	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0

(b) the matrix after using P-M model(50 iterations).

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(c) the matrix after using P-M model(500 iterations).

Figure 4.5: "Block effect" presentation.

Figure 4.5 shows the "block effect" of using the P-M model by giving different numbers of iterations. In order to present this disadvantage more clearly, a usual image denoising procedure by the P-M model is applied and the results including the grey values of a small block which locates at the bottom right corner of number ‘8’ are depicted, from which the "block effect" can also be apparently observed.

(a) the original image

(b) the result after using the P-M model (30 iterations)

95	143	154	147	144	147	133	87	74	76	74	93	157	185
130	175	166	147	152	167	149	88	75	77	74	106	169	187
130	162	119	92	119	157	143	82	75	76	76	120	178	187
128	157	107	74	103	155	139	81	75	76	80	137	182	170
138	158	106	71	106	158	139	82	75	75	89	152	179	142
144	160	139	125	149	174	139	81	75	75	100	165	176	124
141	175	173	167	174	178	138	79	73	77	116	175	169	106
138	170	129	107	132	151	112	76	75	81	131	181	157	89
141	157	94	69	90	92	77	75	74	90	147	182	143	78
141	157	97	72	74	73	76	76	74	99	160	180	130	82
144	155	96	72	75	75	75	75	74	113	171	179	155	144
149	154	95	73	74	76	76	74	78	130	179	187	184	184
157	146	82	72	75	74	76	75	85	144	186	177	147	137
170	125	74	74	75	75	75	73	93	158	185	145	81	70

(c) data of a small block in (a)

104	125	126	128	130	131	132	84	83	83	85	87	155	156
123	127	128	129	131	132	133	85	84	85	86	89	155	156
125	127	128	129	130	132	133	86	85	86	88	100	155	155
127	129	129	129	130	132	133	86	86	86	88	151	153	154
130	131	131	130	131	133	133	87	86	87	88	151	152	152
134	134	134	133	133	133	132	88	87	88	90	150	151	150
136	136	136	135	134	133	131	88	88	89	118	149	149	132
137	137	136	135	133	131	128	88	88	89	133	149	149	129
138	137	90	88	87	87	87	87	87	89	150	152	151	127
138	137	88	87	86	85	85	86	87	89	152	153	152	127
137	136	87	85	84	84	84	85	87	90	153	154	154	154
136	135	86	85	84	84	84	85	86	148	152	154	154	154
135	134	87	85	84	84	84	85	86	149	151	152	154	154
134	132	88	86	85	85	85	86	87	149	150	150	93	93

(d) the result of (c) after 30 iterations

Figure 4.6: "Block effect" presentation by real image denoising application.

4.3.4 Numerical Implementation

In order to dicretise the anisotropic diffusion, (4.13) can be rewritten as

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right). \tag{4.23}$$

The four directional derivatives of u can be defined as below according to Figure 4.7.

$$\nabla_E u_{i,j} = u_{i,j+1} - u_{i,j}, \tag{4.24}$$

$$\nabla_W u_{i,j} = u_{i,j-1} - u_{i,j}, \tag{4.25}$$

$$\nabla_N u_{i,j} = u_{i-1,j} - u_{i,j}, \tag{4.26}$$

$$\nabla_S u_{i,j} = u_{i+1,j} - u_{i,j}. \tag{4.27}$$

Then

$$\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right) = c_E \nabla_E u + c_W \nabla_W u, \tag{4.28}$$

$$\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = c_N \nabla_N u + c_S \nabla_S u, \tag{4.29}$$

where

$$c_N = c(\| \nabla_N u \|),$$

$$c_S = c(\| \nabla_S u \|),$$

$$c_E = c(\| \nabla_E u \|),$$

$$c_W = c(\| \nabla_W u \|).$$

Putting (4.28) and (4.29) together, (4.23) could be discretised as

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t(c_N \nabla_N u + c_S \nabla_S u + c_W \nabla_W u + c_E \nabla_E u) \tag{4.30}$$

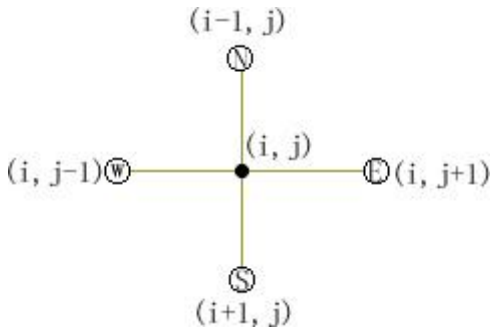


Figure 4.7: Discrete grid of P-M model.

4.3.5 Conclusion

P-M model, as the first nonlinear method by using PDE in image denoising, played a very significant role in the literature. This model makes use of local information and uses an adaptive coefficient to enhance diffusion in smooth areas over areas across edges. Furthermore, the model is easily implemented by using a 4-node neighbourhood to run the iteration. However, as mentioned in previous section, this model can generate the "block effect" phenomenon and cannot remove noisy points with great norm of gradient since the coefficient is small at these points and such that the noise may be kept as edges. In addition, it was shown by Ma îre [85] that coefficients adopted by P-M model make it ill-posed and hence the uniqueness of the solution cannot be guaranteed. Finally, it has no prior knowledge to choose a proper diffusion coefficient, even though some researchers proposed a number of coefficients for the P-M model [81].

4.4 Fourth-Order PDE Model

4.4.1 Motivation and Modelling

As mentioned before, the P-M model promotes strong diffusion in smooth areas and weak diffusion in areas with large norms of gradient. Therefore, as the number of iterations increases, the image will evolve to several regions of which the grey values are the same in each region, i.e., the "block effect". In such a case, the gradient of an arbitrary point in these regions becomes zero. The reason of “block effect” can also be explained from the view of the energy functional. It is shown that the solution of (4.13) is equivalent to the minimisation of an energy functional which is given below when using the diffusivity function of (4.14) [86],

$$J(u) = \int_{\Omega} \frac{K^2}{2} \ln(K^2 + \| \nabla u \|^2) dxdy \tag{4.31}$$

Therefore, $J(u)$ is minimised when $\| \nabla u \|^2$ is minimum, which leads to a piecewise constant approximation of u.

(a) a noisy image (b) the denoised image

Figure 4.8: "Block effect" generated by the P-M model.

To reduce the "block effect" and make a trade-off between noise removal and edge preservation, You and Kaveh developed the fourth-order PDE ([23]) (Y-K model for short) as shown below.

$$\begin{cases} \dfrac{\partial u}{\partial t} = -\nabla^2[c(s)\nabla^2 u], & t > 0,\ (x,y) \subset \Omega \\ \dfrac{\partial u}{\partial n} = 0, & (x,y) \in \partial\Omega \\ u(x,y,0) = u_0(x,y), & t = 0 \end{cases} . \tag{4.32}$$

You and Kaveh chose s as $|\nabla^2 u|$ and used (4.14) as the coefficient attached to $\nabla^2 u$. If the coefficient is chosen as $c(s) = 1/(1+(s/k)^2)$, then the corresponding energy functional can be derived as below,

$$J(u) = \int_\Omega \frac{K^2}{2} \ln(K^2 + |\nabla^2 u|) dxdy \tag{4.33}$$

which employs $|\nabla^2 u|$ instead of $\|\nabla u\|^2$ in (4.31). The minimisation of (4.33) leads to a piecewise planar image, that is, $\nabla^2 u = 0$, locally, as $t \to \infty$. In such case, the Y-K model uses a piecewise planar image to approximate an original clean image. From the perspective of human visualization, a piecewise planar image is more natural than the step images which the second PDEs employ to approximate an original clean image. Take a 1-D signal for instance, both the second-order and fourth-order PDEs provide piecewise lines to approximate the signal. Nevertheless, the line in the second-order PDE model is horizontal

whilst it has a slope in the fourth-order PDE model when $t \to \infty$. Obviously, the approximation obtained by using the fourth-order PDE is more accurate than that obtained by the second-order PDE. This is the reason why it can be used to avoid the "block effect" widely seen in second order PDE models.

4.4.2 Behaviours of Y-K model

In order to understand this model more deeply, the one-dimensional form of (4.32) is taken into consideration ([89]), that is

$$\frac{\partial u}{\partial t} = -\frac{\partial^2}{\partial x^2}\left(c\left(\left(\frac{\partial^2 u}{\partial x^2} \right)^2 \right) \frac{\partial^2 u}{\partial x^2} \right). \tag{4.34}$$

Expanding the right-hand side of (4.34) leads to

$$\frac{\partial u}{\partial t} = -\left(2\left(\frac{\partial^3 u}{\partial x^3} \right)^2 \Phi_1\left((\frac{\partial^2 u}{\partial x^2})^2 \right) \right) \frac{\partial^2 u}{\partial x^2} - \Phi_2\left((\frac{\partial^2 u}{\partial x^2})^2 \right) \frac{\partial^4 u}{\partial x^4}. \tag{4.35}$$

where $\Phi_1(s^2) = 2c''(s^2)s^2 + 3c'(s^2)$ and $\Phi_2(s^2) = 2c'(s^2)s^2 + c(s^2)$. The local behaviour of (4.35) depends on the signs of Φ_1 and Φ_2. If $\Phi_1\left((\partial_x^2 u)^2\right) < 0$, it leads to the second order forward diffusion, otherwise the second order backward diffusion. Likewise, if $\Phi_2\left((\partial_x^2 u)^2\right) > 0$, the equation performs the fourth-order forward diffusion, whereas $\Phi_2\left((\partial_x^2 u)^2\right) < 0$ ensures the fourth order backward diffusion.

4.4.3 Numerical Implementation

(4.32) can be solved by means of an iterative method. Rewrite (4.32) as:

$$\frac{\partial u}{\partial t} = -\nabla^2 g,$$

$$g = c(\nabla^2 u\,|)\nabla^2 u.$$

The discretization process can be applied as following steps.

Step 1: Calculate $\nabla^2 u$ as shown in (4.10).

$$\nabla^2 u_{i,j}^n = u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n \quad i = 0,1,2,\cdots I,\ j = 0,1,2,\cdots J,$$

With the symmetric boundary conditions

$$\begin{matrix} u_{-1,j}^n = u_{0,j}^n, u_{I+1,j}^n = u_{I,j}^n \\ u_{i,-1}^n = u_{i,0}^n, u_{i,J+1}^n = u_{i,J}^n \end{matrix}.$$

Step 2: Calculate function *g*

$$g_{i,j}^n = c(\nabla^2 u_{i,j}^n)\nabla^2 u_{i,j}^n.$$

Step 3: Calculate $\nabla^2 g$

$$\nabla^2 g_{i,j}^n = \frac{g_{i+1,j}^n + g_{i-1,j}^n + g_{i,j+1}^n + g_{i,j-1}^n - 4g_{i,j}^n}{h^2},\ i = 0,1,2,\cdots I,\ \ j = 0,1,2,\cdots J,$$

with symmetric boundary conditions

$$\begin{matrix} g_{-1,j}^n = g_{0,j}^n, g_{I+1,j}^n = g_{I,j}^n \\ g_{i,-1}^n = g_{i,0}^n, g_{i,J+1}^n = g_{i,J}^n \end{matrix}.$$

Step 4: Calculate the iterative equation

$$u_{i,j}^{n+1} = u_{i,j}^n - \Delta t \nabla^2 g_{i,j}^n.$$

Step 5: Go to **Step 1** if the pre-assigned number of iteration is not completed.

4.4.4 Conclusion

The drawback of the Y-K model is that it produces some black or white isolated speckles. These speckles can be featured as the pixels which are much darker or lighter than their adjacent pixels. You and Kaveh thought that the piecewise planar images have weaker masking capability than step images used in anisotropic diffusion. To avoid this, You and Kaveh gave a revised resolution by taking the medium filtering on the denoised image by the Y-K model. Although it can remove the isolated speckles, the image becomes blurred after medium filtering. Therefore it does not settle the drawback of the Y-K model.

4.5 Closure

In this chapter, the three typical PDE-based methods from the literature were carefully examined. The first model is the isotropic diffusion model which is the same as Gaussian filtering. The second model is the anisotropic diffusion model (the P-M model) which uses a second order PDE to remove the noise from an image like a diffusion procedure. The last is a fourth order PDE model (the Y-K model) which can optimise the trade-off between denoising and preserving edges. Advantages and disadvantages of these models were also discussed and several improved models will be described in the next chapters.

Chapter 5. MODIFIED PDE MODELS FOR IMAGE DENOISING

In this chapter, some novel improved PDE models are proposed to overcome the drawbacks in the P-M model and the Y-K model. Section 5.1 analyses further behaviours of the P-M model and proposes a new choice of coefficient for it. In Section 5.2, a new weighted sum operator, which takes advantage of both the gradient and Laplacian operators of an image, is introduced to overcome the drawback of the Y-K model. In the last section, an adaptive relaxation method with discontinuity treatment is employed to reduce the speckles generated by the Y-K model. Furthermore, a control of the iterative process for fourth-order PDE models is also included to provide an iterative control.

5.1 A Modified Coefficient for the Anisotropic Model (IPM)

5.1.1 Model description and analysis

The choice of diffusion coefficient in the P-M model is a vital step for image denoising [86]. A good choice prevents diffusion across edges and allows diffusion in smooth regions of the image. Different choices of the diffusion coefficient may lead to entirely distinctive results. Besides the coefficients suggested by Perona and Malik, various different types of diffusion coefficients providing good results were proposed in [90] - [92]. According to the work by You [88], the P-M model is well-posed if and only if the function ϕ defined in (4.16) which is

$$\phi(s) = sc(s), \tag{5.1}$$

satisfies the derivative condition

$$\phi'(s) \geq 0 \; for \; s \geq 0. \tag{5.2}$$

Well-posedness means a unique solution of the PDE exists and continuously depends on the initial data, in other words, the solution is stable. Otherwise it is ill-posed. Consider the one dimensional P-M model,

$$u_t - \phi'(|u_x|)u_{xx} = 0, \tag{5.3}$$

if $\phi(s)$ is decreasing, i.e. $\phi'(s) < 0$, at some point s and if there is a certain point x, subject to the condition $\|u'(x)\| = s$, one obtains a negative derivative, say $-k$ (k is a positive number), at the point x, so that (5.3) becomes

$$u_t + ku_{xx} = 0. \tag{5.4}$$

(5.4) is a backward heat equation which is well known to be an ill-posed problem in physics [93]. Therefore, for functions $c(s)$ with non-increasing $\phi(s)$, a non-deterministic and unstable process may happen, which means that slightly perturbed images might produce very different edges and solutions.

According to the above description and in order to take care of the well-posedness, a new coefficient is proposed and analysed. This new coefficient is defined as

$$c(s) = s^{p-2}, 1 < p < +\infty. \tag{5.5}$$

It is easy to verify that

$$\phi'(s) = (p-1)s^{p-2} \geq 0, \ \forall p > 1 \text{ and } s \geq 0$$

Figure 5.1 shows the shape of (5.5) when p = 1.3.

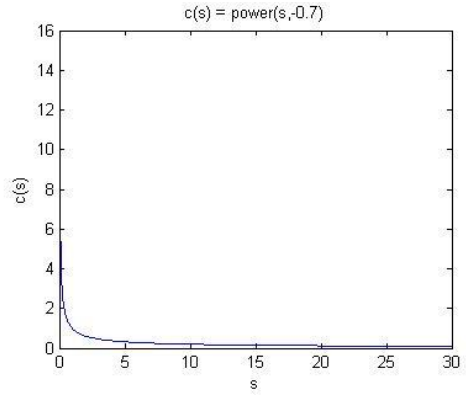


Figure 5.1: The shape of $c(s) = s^{p-2}, p = 1.3.$

Then the Improved P-M model (IPM) can be written as

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla \bullet (\| \nabla u \|^{p-2} \nabla u), & t > 0,\ (x, y) \in \Omega \\ \dfrac{\partial u}{\partial n} = 0, & (x, y) \in \partial\Omega \\ u(x, y, 0) = u_0(x, y), & t = 0 \end{cases} . \tag{5.6}$$

In order to analyse the new improved coefficient and understand how the anisotropic diffusion works locally, the Gauge coordinate system as shown in Figure 5.2 is adopted.

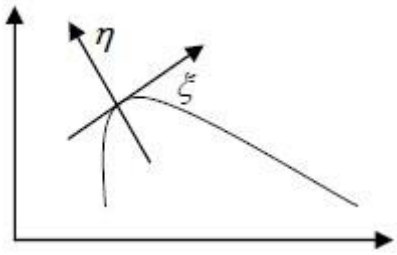


Figure 5.2:The Gauge coordinate System.

The two orthogonal unit vectors η and ξ represent the gradient vectors alone the direction of maximal change of intensity, and the corresponding perpendicular vector tangential to the isophote respectively. Generally, these two vectors can be written as

$$\eta = \frac{(u_x, u_y)}{\sqrt{u_x^2 + u_y^2}}, \xi = \frac{(-u_y, u_x)}{\sqrt{u_x^2 + u_y^2}}.$$

By using the Hessian matrix of the original coordinates,

$$H = \begin{pmatrix} u_{xx} & u_{xy} \\ u_{yx} & u_{yy} \end{pmatrix},$$

arbitrary second-order directional derivatives can be calculated by,

$$u_{\xi\eta} = \xi' H \eta \tag{5.7}$$

With (5.7), the second-order derivatives of *u* with respect to η and ξ are show as below.

$$u_{\eta\eta} = \frac{u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}}{u_x^2 + u_y^2},$$

$$u_{\xi\xi} = \frac{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}{u_x^2 + u_y^2}.$$

It can be easily seen that

$$u_{xx} + u_{yy} = u_{\xi\xi} + u_{\eta\eta}.$$

By using the Gauge coordinate system and making the following algebraic steps

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \nabla\bullet(c(\|\nabla u\|)\nabla u) \\
&= c(\|\nabla u\|)\nabla^2 u + \nabla u \nabla c(\|\nabla u\|) \\
&= c(\|\nabla u\|)(u_{xx}+u_{yy}) + u_x \frac{\partial c(\|\nabla u\|)}{\partial x} + u_y \frac{\partial c(\|\nabla u\|)}{\partial y} \\
&= c(\|\nabla u\|)(u_{xx}+u_{yy}) + u_x c'(\|\nabla u\|)\frac{u_x u_{xx}+u_y u_{xy}}{\sqrt{I_x^2+I_y^2}} + u_y c'(\|\nabla u\|)\frac{u_y u_{yy}+u_x u_{xy}}{\sqrt{u_x^2+u_y^2}} \\
&= c(\|\nabla u\|)(u_{\xi\xi}+u_{\eta\eta}) + c'(\|\nabla u\|)\frac{u_x^2 u_{xx}+2u_x u_y u_{xy}+u_y^2 u_{yy}}{\sqrt{u_x^2+u_y^2}} \\
&= c(\|\nabla u\|)(u_{\xi\xi}+u_{\eta\eta}+\frac{c'(\|\nabla u\|)\|\nabla u\|}{c(\|\nabla u\|)}u_{\eta\eta}) \\
&= c(\|\nabla u\|)u_{\xi\xi} + c(\|\nabla u\|)(1+\frac{c'(\|\nabla u\|)\|\nabla u\|}{c(\|\nabla u\|)})u_{\eta\eta}.
\end{aligned}$$

The anisotropic diffusion can be rewritten as

$$\frac{\partial u}{\partial t} = c(\|\nabla u\|)u_{\xi\xi} + c(\|\nabla u\|)(1+\frac{c'(\|\nabla u\|)\|\nabla u\|}{c(\|\nabla u\|)})u_{\eta\eta}. \tag{5.8}$$

Substituting (5.5) into (5.8), the equation becomes

$$\frac{\partial u}{\partial t} = \|\nabla u\|^{p-2} u_{\xi\xi} + (p-1)\|\nabla u\|^{p-2} u_{\eta\eta}. \tag{5.9}$$

(5.9) is a nonlinear anisotropic diffusion equation. The diffusion coefficients $\|\nabla u\|^{p-2}$ and $(p-1)\|\nabla u\|^{p-2}$ control the diffusion in the directions of ξ and η respectively. The first term of (5.9) represents a degenerated forward diffusion in the direction orthogonal to the gradient. Thus, this directional smoothing process should be encouraged since it represents a well-posed smoothing operation that could preserve edges since an edge is also orthogonal to the gradient. The second term is always non-negative which ensures that the PDE is well-posed. In other words, the value of p is required to be chosen properly so that the diffusion results become meaningful in image processing.

There are two special cases as listed below,

(1) For $p \to 1$, (5.6) degenerates to the TV model [19] which is diffusive only along the direction tangential to the gradient.

(2) For $p \to 2$, (5.6) becomes the isotropic diffusion model as shown in (4.5).

5.1.2 Numerical implementation

In order to avoid $\| \nabla u \| = 0$ which makes the algorithm unstable, a small positive regularisation parameter ε is used, i.e., $\| \nabla u \|_\varepsilon = \sqrt{\varepsilon^2 + \| \nabla u \|^2}$, (5.6) can be discretised as,

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t \cdot \lambda \cdot (c_N \cdot \nabla_N u_{i,j} + c_S \cdot \nabla_S u_{i,j} + c_E \cdot \nabla_E u_{i,j} + c_W \cdot \nabla_W u_{i,j}),$$

where

$$\lambda \in [0, 1/4]$$

$$\nabla_N u_{i,j} = (u_{i-1,j} - u_{i,j}) / h,$$

$$\nabla_S u_{i,j} = (u_{i+1,j} - u_{i,j}) / h,$$

$$\nabla_E u_{i,j} = (u_{i,j+1} - u_{i,j}) / h,$$

$$\nabla_W u_{i,j} = (u_{i,j-1} - u_{i,j}) / h,$$

$$c_N = c(\| \nabla_N u \|_\varepsilon),$$

$$c_S = c(\| \nabla_S u \|_\varepsilon),$$

$$c_E = c(\| \nabla_E u \|_\varepsilon),$$

$$c_W = c(\| \nabla_W u \|_\varepsilon).$$

In the numerical experiments, the values of λ, Δt , and h were chosen as $\frac{1}{4}$, 1, and 1 respectively.

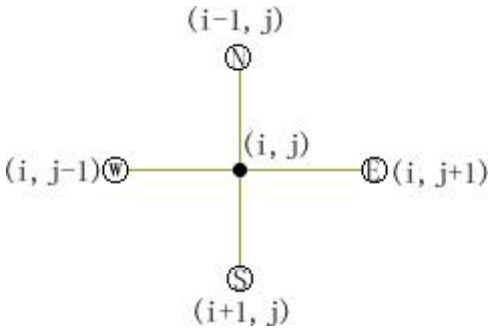


Figure 5.3: The mesh setup of IPM model.

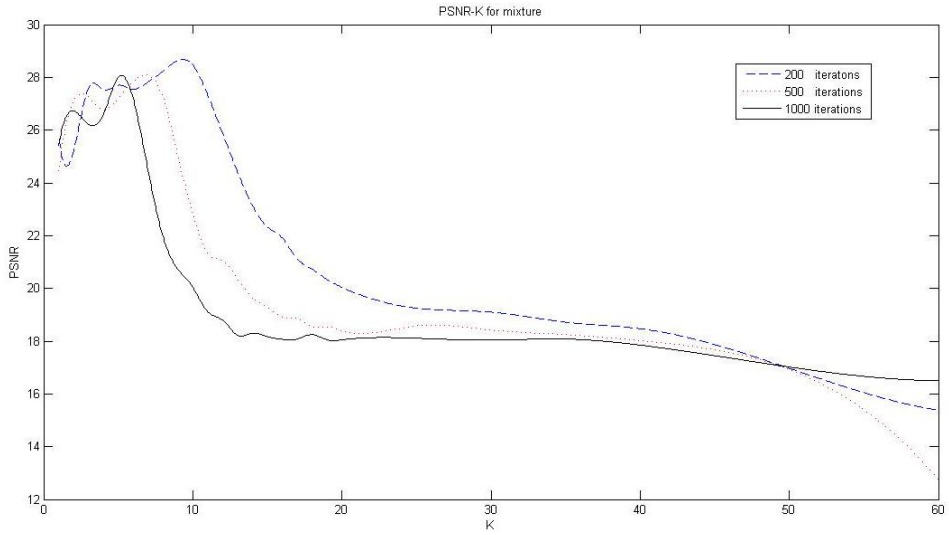
5.2 A Weighted Sum Operator for the Fourth-Order PDE Model

5.2.1 Choice of *K*

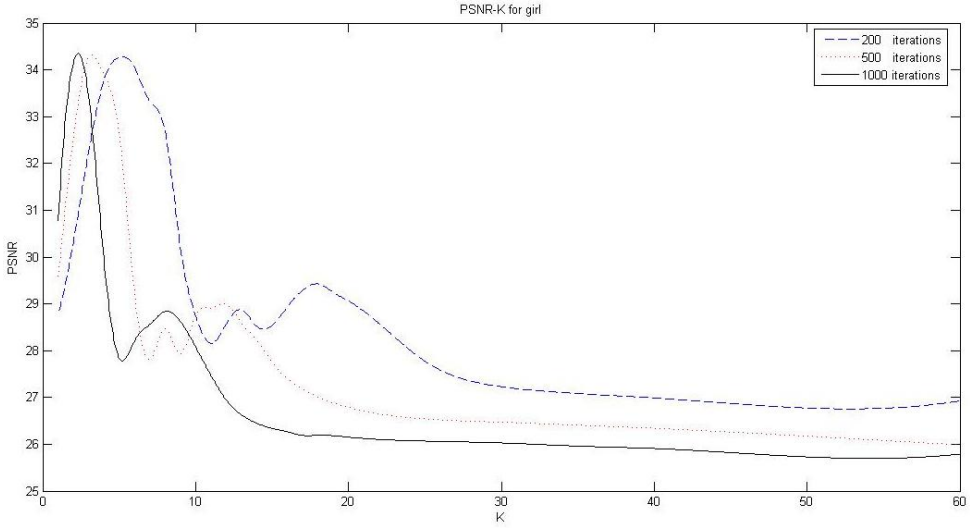
The parameter *K*, in the coefficient $c(s)$, plays a significant role in the denoising procedure. Different values of *K* would lead to different results [12], [14], [87]. Although in the literature, research has been conducted in the choice of K in the P-M model and some good results were obtained, the performances of the Y-K model and the P-M model are different in such a way that values of K used for the P-M model may not be suitable for the Y-K model. Therefore, some experiments are provided below in order to examine possible appropriate values of K in the Y-K model. Figure 5.4 (a) is a designed image which includes different types of edges, such as horizontal, vertical, and slope edges. Figure 5.4 (c) is a popular image in image processing area. Figure 5.4 (b) and (d) are the two corresponding images with Gaussian noise.

(a) mixture (b) mixture with Gaussian noise (c) girl (d) girl with Gaussian noise

Figure 5.4: Two sets of testing images.



(a) mixture



(b) girl

Figure 5.5: PSNR-K profile with 200,500, and 1000 iterations.

From these two graphs in Figure 5.5, it can be observed that with different iterations, the value of PSNR (The definition of PSNR is discussed in Chapter 7) reaches its peak value at different values of K and furthermore, at all cases, the values of K which achieve greater values of PSNR range between 0 and 10. Therefore, the value of K may be possible related to the iteration and bounded at $(0,10]$.

5.2.2 The Drawback of Y-K Model

The replacement of the gradient operation by the Laplacian operator in Y-K model makes use of those trivial Laplacian points within image support for planar and neighbourhood of those

points. Hence, the "block effect" can be avoided by using piecewise slope planes to approximate the image without noise. However, this method generates some isolated speckles. You and Kaveh considered it as that the piecewise planar image has less masking capability than the step image which is used to approximate the original image in the anisotropic diffusion model. An example of the isolated speckles generated by Y-K model is shown in Figure 5.7.

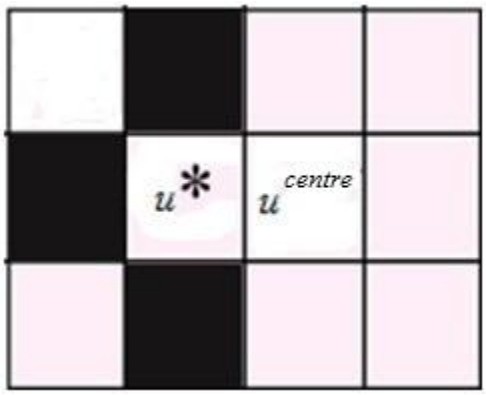


Figure 5.6: A typical structure generating isolated point.

Consider a pixel denoted by $u*$ with a grey value near 255 (white colour) and located near an edge, as shown in Figure 5.6. If the edge is almost black, the value of $|\nabla^2 u^*|$ could be large. From Figure 4.4(b) the value of $c(|\nabla^2 u^*|)\nabla^2 u^*$ is small and negative. However, the absolute value of the Laplacian at the centre point (denoted by u^{centre}) $|\nabla^2 u^{centre}|$ is very small but $c(|\nabla^2 u^{centre}|)\nabla^2 u^{centre}$ is relatively large. In the case of $c(|\nabla^2 u^{centre}|)\nabla^2 u^{centre} > 0$, the difference between u^{centre} and $u*$ is larger such that u^{centre} is sharpened and is brighter than its neighbourhood. Likewise, the reason for generating the black points is similar. In other words, an improper choice of $c(\cdot)$ and s may leave certain brighter points in dark areas (See Figure 5.7(b)) and darker points in brighter areas (See Figure 5.7 (c)).

(a) the denoised image

(b) zoom-in to white points

(c) zoom-in to black points

Figure 5.7: Isolated points generated by Y-K model.

5.2.3 The Weighted Sum Model

To avoid the isolated points generated by Y-K model, a modification is proposed in this thesis. In essence the term $|\nabla^2 u|$ used in $c(|\nabla^2 u|)$ is the main reason for causing isolated points, thus the coefficient c should not depend solely on $|\nabla^2 u|$. Since $\|\nabla u\|$ represents the edge information, it would be sensible to include the gradient term into the variable s. Therefore the following weighted sum

$$s = \omega_1 \nabla^2 u + \omega_2 \|\nabla u\|, \tag{5.10}$$

may be used to handle the generation of isolated points. Here ω_1 and ω_2 are the weighted coefficients satisfying $\omega_1 + \omega_2 = 1$. There are different choices between edge preserving and isolated points removal by changing the values of ω_1 and ω_2. The model which replaces (4.32) becomes

$$\begin{cases} \dfrac{\partial u}{\partial t} = -\nabla^2 [c(\omega_1 \nabla^2 u + \omega_2 \|\nabla u\|)\nabla^2 u], & t > 0,\ (x,y) \in \Omega \\ \dfrac{\partial u}{\partial n} = 0, & (x,y) \in \partial\Omega \\ u(x,y,0) = u_0(x,y), & t = 0 \end{cases} . \tag{5.11}$$

(5.11) is named as the weighted Y-K (WYK) model.

There are two special cases of the WYK model. The first case is when $\omega_1 = 1$ and $\omega_2 = 0$ where (5.11) becomes the Y-K model. The second case is that when $\omega_1 = 0$ and $\omega_2 = 1$ where

the fourth order PDE couples with the edge stopping function of the P-M model. This reduces the diffusion according to the gradient instead of the Laplacian of grey values.

In order to ensure the fourth-order PDE is well-posed ([88]), i.e.

$$\phi'(s) \geq 0,$$

where $\phi(s)$ is defined in (4.16), a new improved coefficient is adopted to replace (4.14)([94]) as follows.

$$c(s) = \frac{1}{\sqrt{1+(\frac{s}{K})^2}},$$

where K is a positive constant. It can be shown that

$$\phi'(s) = (1+\frac{s}{K})^{-\frac{3}{2}} \geq 0,$$

i.e. it satisfies the well-posed condition. The improved coefficient aims to avoid the isolated points brought in by the Y-K model.

5.2.4 Numerical Implementation

(5.11) can be solved by means of an iterative method. Rewrite (5.11) as:

$$\frac{\partial u}{\partial t} = -\nabla^2 g.$$

$$g = c(\omega_1 \nabla^2 u + \omega_2 \| \nabla u \|) \nabla^2 u.$$

The iterative process is listed below.

Step 1: Calculate $\nabla^2 u$ and $\| \nabla u \|$

$$\nabla^2 u_{i,j}^n = \frac{u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n}{h^2}.$$

$$\| \nabla u_{i,j}^n \| = \left[\left(\frac{u_{i+1,j}^n - u_{i,j}^n}{h} \right)^2 + \left(\frac{u_{i,j+1}^n - u_{i,j}^n}{h} \right)^2 \right]^{\frac{1}{2}}, \; i = 0,1,2,\cdots I, \; j = 0,1,2,\cdots J.$$

with the symmetric boundary conditions

$$u^n_{-1,j} = u^n_{0,j}, u^n_{I+1,j} = u^n_{I,j},$$
$$u^n_{i,-1} = u^n_{i,0}, u^n_{i,J+1} = u^n_{i,J}.$$

Step 2: Calculate function g

$$g^n_{i,j} = c(\omega_1 \| \nabla u^n_{i,j} \| + \omega_2 \nabla^2 u^n_{i,j}) \nabla^2 u^n_{i,j}.$$

Step 3: Calculate $\nabla^2 g$

$$\nabla^2 g^n_{i,j} = \frac{g^n_{i+1,j} + g^n_{i-1,j} + g^n_{i,j+1} + g^n_{i,j-1} - 4g^n_{i,j}}{h^2},\ i = 0,1,2,\cdots I,\ \ j = 0,1,2,\cdots J.$$

with the symmetric boundary conditions

$$g^n_{-1,j} = g^n_{0,j}, g^n_{I+1,j} = g^n_{I,j}$$
$$g^n_{i,-1} = g^n_{i,0}, g^n_{i,J+1} = g^n_{i,J}$$

Step 4: Calculate the iterative equation

$$u^{n+1}_{i,j} = u^n_{i,j} - \Delta t \nabla^2 g^n_{i,j}.$$

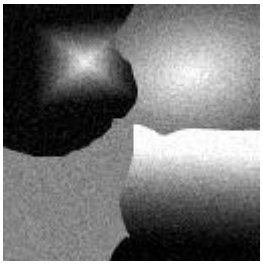
Step 5: Go to **Step 1** if the pre-assigned number of iteration is not completed.

Here, $\omega_1 = 0.9, \omega_2 = 0.1.$

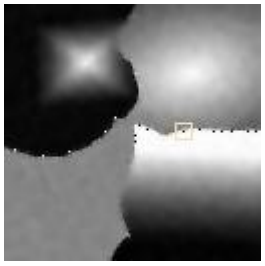
5.3 An Adaptive Relaxation Method with Discontinuity Treatment for the Fourth-Order PDE Model (AYK Model)

5.3.1 More about the Y-K model

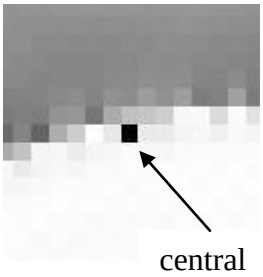
Reasons of isolated speckles created by using the Y-K model were discussed in section 5.2.2. In order to have a better understanding of the creation of these speckles, a controlled process is applied to the black point (denoted as central points) as labelled in Figure 5.8 and its four neighbouring points.



(a) a noisy image



(b) the denoised image



(c) zoom-in of the square part of (b)

Figure 5.8: Isolated speckles in the denoised image by the Y-K model.

The changes of the intensity values on the central point and its four neighbourhoods are shown in Figure 5.9:

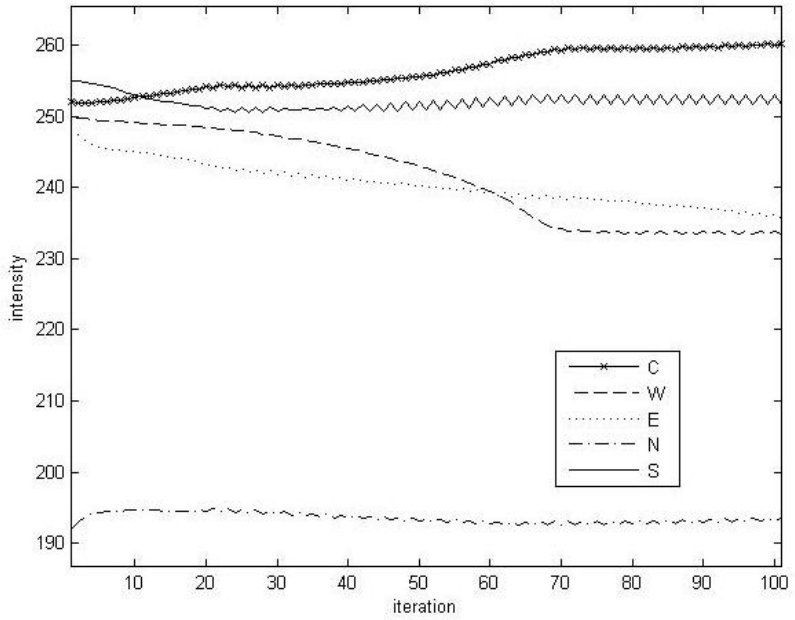


Figure 5.9: Change of the intensity around isolated speckles.

The intensity of the central point increases whereas the intensities of the neighbouring points either decrease or fluctuate slightly. Thus, the difference of intensities between the central point and its neighbouring points becomes bigger and bigger as the iteration goes on. On the other hand, from the changes of intensity values of the central point, it is noticed that the value of intensity is above the extreme value 255 when storing an image. Thus, if the image is being stored at this current iteration, the particular central point would be returned back to a black point. The same case applies to those white points. From the above discussion, the reasons of isolated speckles can be summarised as

1) The intensity of some points is changing quicker than those of their neighbouring points and they change in different ways, i.e. the intensity of central point increases and those of its neighbouring points decrease.
2) The intensities of some points are out of the range [0, 255] for a grey-value image used in computer implementation.

5.3.2 An Adaptive Relaxation Method for the Y-K model

Relaxation methods are established in numerical schemes in many areas involving the solutions of simultaneous equations from discretisation schemes. These simultaneous equations may be linear or nonlinear. An iterative scheme incorporating relaxation parameters supported with an initial appoximation is constructed and obtains an improved approximation such that the error is reduced until it is less than some specified tolerance [95].

In general, relaxation methods are used to control the variation of approximate solutions between consecutive iterations. As mentioned in section 5.3.1, speckles appear in the numerical solution process because pixel intensities of some points are changing faster than other points. Therefore, it is reasonable to employ a relaxation method in a denoising process when using the Y-K model. In this section, an adaptive relaxation method is proposed aiming to relieve the generation of speckles when using the Y-K model. In order to identify and detect isolated speckles that have attained their extremum pixel values within a 3×3 neighbourhood window, and isolated-point-detection scheme is constructed in section 5.3.3. Such isolated-point-detection scheme is added into the algorithm with a relaxation method. The adaptive relaxation method can be explained as follows.

Suppose u^n is the iterative solution of the discrete approximation of (4.32) at time $t = n\Delta t$, where Δt is the time step size of the iteration, for a point $p \in \Omega$, u_{max} and u_{min} are the maximum and minimum intensity values of a deleted neighbourhood of p (i.e. neighbourhood of p without p). Let $u_{globalmax}$ and $u_{globalmin}$ be the global maximum and minimum value of the image u. Then define $globalmax = 0.9 \cdot u_{globalmax}$ and $globalmin = 1.1 \cdot u_{globalmin}$. If $u_p^n \notin [u_{max}, u_{min}]$ or $u_p^n \notin [globalmin, globalmax]$, then one has

$$u_p^n = (1 - \lambda(n))u_p^{n-1} + \lambda(n)u_p^n \tag{5.12}$$

Here $\lambda:[0,+\infty]\mapsto[0,1]$ is a monotonically non-decreasing function with respect to the number of iterations. The function of λ used in this thesis is defined as below.

$$\lambda(n)=1-e^{(-0.01n)} \tag{5.13}$$

If the intensity of the central point in a 3 × 3 window changes significantly, it can be restrained by using (5.12) to avoid a rapid variation of the solution. The result obtained by the adaptive relaxation method for Figure 5.8(a) is shown in Figure 5.10(a). It can be seen from the result that the isolated speckles are diminished. However, it seems that the intensities along edges are not continuous. Therefore, a discontinuity treatment for edges is imposed on those points in section 5.3.3. The aim of this treatment is to make the edges look more natural in an image following the denoising procedure.

(a) (b)

Figure 5.10: Results of using relaxation methods. (a) and (b) are the denoised images by the Y-K model without and with discontinuity treatment.

5.3.3 The Discontinuity Treatment

To handle the discontinuity on the edges, the first step is to detect the discontinuous pixels along the edges. In the literature, the mask below is used to recognise the discontinuous points around the edges,

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix},$$

The response, R, of the mask applied at any point in an image is given by

$$R=\sum_{i=1}^{9} w_i u_i\,, \tag{5.14}$$

where u_i is the intensity of the pixel at position i in the mask above and w_i are the weights of the mask. Detection of the discontinuous points on which the mask is centered occurs if

$$|R| \geq T . \tag{5.15}$$

Here T is a non-negative threshold. The underlying idea of this method is to make use of the intensity difference, which is determined by the threshold T, between an isolated point and its neighbouring points.

After detecting the discontinuous points on the edges, in order to restore better values of the pixels at such points, the types of edges where these points are located need to be examined. For simplicity, only four simple types of edges are to be taken into consideration, including horizontal edges, vertical edges, and inclined edges oriented at 45^{o} and -45^{o} directions. The corresponding masks are shown as below [96]:

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

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

Figure 5.11: Edge detector masks corresponding to the horizontal, vertical, 45° and -45° directions.

Four different masks are proposed here to restore the discontinuous pixel points on edges defined in the above modes respectively (See Figure 5.12). By considering that the intensity of the central point should contribute itself the most than the neighbouring points and points along the same edges should have more similarity, here the weighted coefficients are chosen such that the central point takes the most weight and then the points along the edge on which the central point is located at. Other points take less weight. For example, if a discontinuous pixel is located on a horizontal edge, then the template to be used is as in Figure 5.12(a). The result of using the relaxation method together with the discontinuity treatment is shown in Figure 5.10(b), which shows the performance of this discontinuity treatment.

$$\begin{bmatrix} 0.05 & 0.05 & 0.05 \\ 0.2 & 0.3 & 0.2 \\ 0.05 & 0.05 & 0.05 \end{bmatrix} \quad \begin{bmatrix} 0.05 & 0.2 & 0.05 \\ 0.05 & 0.3 & 0.05 \\ 0.05 & 0.2 & 0.05 \end{bmatrix} \quad \begin{bmatrix} 0.05 & 0.05 & 0.2 \\ 0.05 & 0.3 & 0.05 \\ 0.2 & 0.05 & 0.05 \end{bmatrix} \quad \begin{bmatrix} 0.2 & 0.05 & 0.05 \\ 0.05 & 0.3 & 0.05 \\ 0.05 & 0.05 & 0.2 \end{bmatrix}$$

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

Figure 5.12: Discontinuous point restored masks corresponding to the horizontal, vertical, 45 °and -45° directions.

In order to present the performance of the proposed method, a set of images with different simple edges are designed to test the Y-K model and the proposed model when coming across the discontinuity at the edges.

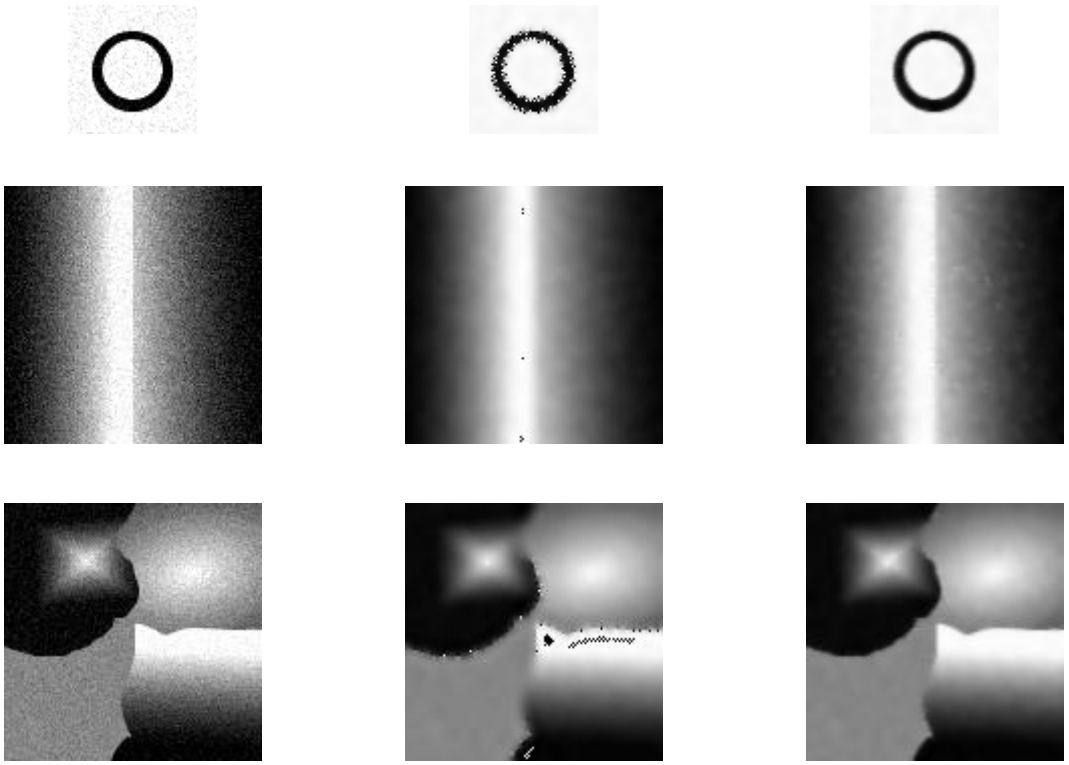


Figure 5.13: A set of experiments by using the Y-K model and the proposed model. Column (a): Three test images with 10 dB Gaussian noise, Column (b): The corresponding denoised images by the Y-K model, Column (c): The corresponding denoised images by the proposed model.

From Figure 5.13, it can be seen that no matter what kind of edges are present, the Y-K model cannot handle them well and isolated speckles exist both in the flat area near the edges and on the edges. Especially for the circle image, after processing by the Y-K model, although the noise in the flat area is removed, there are many speckles generated around the edges. However, the resulting images in Figure 5.13(c) show that the proposed model not only removes noise efficiently, but also succeeds in avoiding the speckles and preserves edges better than the Y-K model.

5.3.4 A Control of Iteration

In numerical analysis, the L_2-norm is often used to control the convergence, that is ([97]),

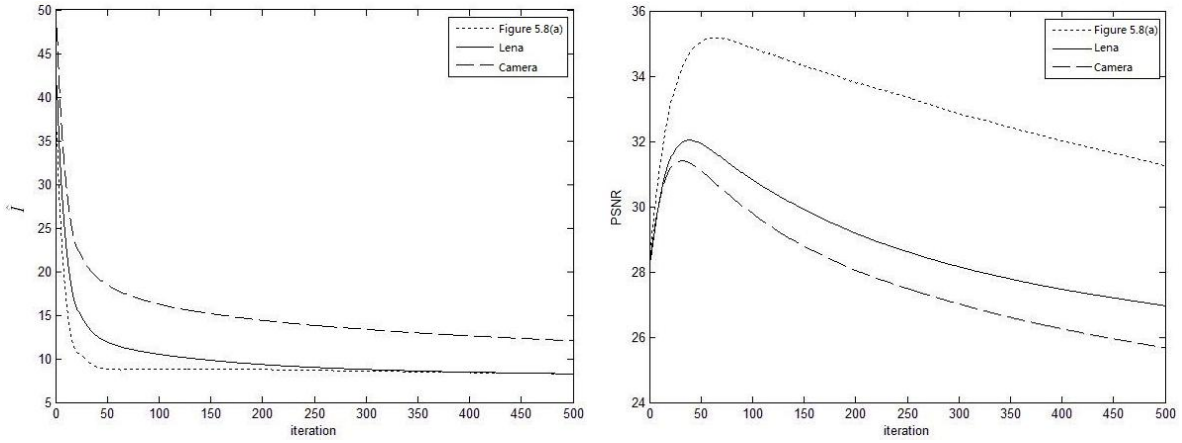
$$\left\| u^n - u^{n-1} \right\|_{L_2} = \sqrt{\sum_{i,j \in \Omega} (u_{i,j}^n - u_{i,j}^{n-1})^2} < \varepsilon \tag{5.16}$$

where Ω is the problem domain and ε is the tolerance usually set as 10^{-4}. However, this norm is not usually employed in image processing. In the literature, most papers simply set the number of iterations as an input [4], [10], [12], and some papers proposed different criteria according to their specific models [98]. Therefore, it is sensible to find a certain quantity to control the number of iterations. As mentioned in section 4.4, with the iteration proceeding, the asymptotic value of $\nabla^2 u$ lies close to zero, as $t \to \infty$, which means $\nabla^2 u$ could be used to terminate the iteration. In this section, the average value of $\nabla^2 u$ over all pixels is proposed to control the iteration process.

$$\hat{I} = \frac{\sum_{i,j} | \nabla^2 u_{ij} |}{M \times N}, \tag{5.17}$$

where M and N are the size of an image.

In order to investigate the features of $\hat{I}$, the relationship between $\hat{I}$ and the number of iterations is studied using the Y-K model. Here the image in Figure 5.8(a) and two other benchmarking images in Figure 7.2 are used for testing.



(a) (b)

Figure 5.14: Profiles of $\hat{I}$ and PSNR with regard to the number of iterations.

Figure 5.14(a) shows that after certain number of iterations, the value of $\hat{I}$ becomes constant. Furthermore, by comparing Figure 5.14(a) and Figure 5.14(b), it is easily obtained that when the values of PSNR reach their peak, values of $\hat{I}$ tend to become constant. Therefore, it is reasonable to assume that when the value of $\hat{I}$ is nearly constant, the value of PSNR is most likely to be high. Based on this analysis, the following condition is proposed to control the iteration:

$$\left|\hat{I}^n - \hat{I}^{n-1}\right| < \varepsilon \tag{5.18}$$

Here ε is a tolerance which can be chosen for different applications.

5.3.5 Numerical Implementation

The discretised scheme is the same as that in section 6.1.2. With this scheme, the improved algorithm can be described as below.

1. Input an image u;

2. Calculate $\nabla^2 u$ and $\nabla^2 g = \nabla^2\left(c(\nabla^2 u)\nabla^2 u\right)$;

3. Calculate $u^{n+1} = u^n - \Delta t \nabla^2 g^n$;

4. Apply the relaxation method to those points which change too fast comparing with other points: If $u^n \notin \left[u_{min}, u_{max}\right]$ or $u^n \notin [globalmin, globalmax]$, $u^{n+1} = \left(1-\lambda(n)\right)u^{n+1} + \lambda(n)u^n$;

5. Check the stopping creterion: If $\left|\hat{I}^n - \hat{I}^{n-1}\right| \geq \varepsilon$, goto 2, else goto 6;

6. Use the discontinuity treatment in section 6.3.3 to restore the discontinous points on the edges.

Remarks:

a: The discretised form of Laplacian is computed as below:

$$\nabla^2 u_{i,j}^n = \frac{u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n}{h^2},\ i = 0,1,2,\cdots I,\ j = 0,1,2,\cdots J.\ ;$$

b: The symmetric condition is used to deal with the boundary:

$$\begin{aligned} u_{-1,j}^n &= u_{0,j}^n, u_{I+1,j}^n = u_{I,j}^n \\ u_{i,-1}^n &= u_{i,0}^n, u_{i,J+1}^n = u_{i,J}^n \end{aligned}.$$

c: u_{max} and u_{min} are the local maximum and minimum values of a 3 by 3 neighbourhood in the current point;

d: *globalmax* and *globalmin* are the 90% of the global maximum and 110% of the global minimum values in the current image;

e: $\varepsilon = 10^{-3}$;

f: the mask used in (5.14) is

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

g: (5.15) is modified as $|R| > Max$ or $|R| < Min$ and the values of *Max* and *Min* are the same as those in Remark *c*;

Chapter 6. COUPLING THE ACTIVE CONTOUR MODEL AND THE ANISOTROPIC DIFFUSION MODEL

6.1 Active Contour Model

The active contour model is a model related to the evolution of a two dimensional dynamic closed curve. The curve moves towards the edges of interest under the combined contribution of internal force and external force. It is a mechanism of introducing a certain degree of prior knowledge into the low-level image interpretation. The active contour model was initially proposed by Kass [31] and is known as the snake model. It provides a unified solution for various problems in computer vision. In recent years, it has been applied in many areas in computer vision, such as edge detection [99], 3D reconstruction [100], and stereo vision matching [101].

6.1.1 Model Description

Given an image u and a closed curve $v(s) = (x(s), y(s)), s \in [0,1]$, the total energy in the snake model is defined as

$$J_{snake}(\mathbf{v}(s)) = \int_0^1 E_{int}\left(\mathbf{v}(s)\right) + E_{ext}\left(\mathbf{v}(s)\right) ds\,, \tag{6.1}$$

where $E_{int}\left(\mathbf{v}(s)\right)$ and $E_{ext}\left(\mathbf{v}(s)\right)$ are the internal and external energy terms respectively. The internal energy is the contour influence which is used to control the geometry of the curve locally. It is related to the elasticity and stiffness of the curve. The external energy, including

the image energy and constraints defined by the users, is meant to drive the curve towards the boundary of the object according to some prior knowledge.

The internal energy proposed by Kass is defined as

$$E_{int}(\mathbf{v}(s)) = \alpha(s)\left\|\mathbf{v}'(s)\right\|^2 + \beta(s)\left\|\mathbf{v}''(s)\right\|^2. \tag{6.2}$$

Here, $\mathbf{v}'(s)$ is the first derivative of the curve. The integral of it measures the Euclidean length of the curve. This term is similar with the elastic potential energy of a string [103], which makes the curve behave like an elastic string. This means that when minimising the integral of this term, the length of the closed curve tends to be as small as possible, i.e. the curve shrinks. $\mathbf{v}''(s)$ is the second derivative of the curve known as the curvature term which is used to control the smoothness of the curve. In other words, the two terms in (6.2) are the measurement of elasticity and rigidity of the curve respectively. The coefficients $\alpha(s)$ and $\beta(s)$ are used to control the influence of elasticity and stiffness terms. Minimisation of the integral of (6.2) means that the curve would slither towards the centre of the closed area and consequently become a point as shown in Figure 6.1.

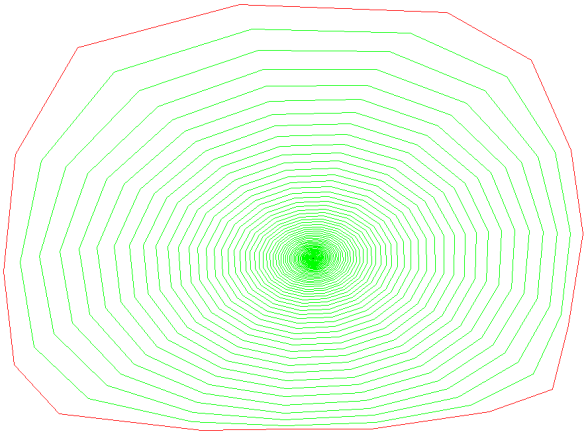


Figure 6.1: Snake Demonstration. The outer black curves are the initial curves and the centre point is the final 'curve'. Green curves are the traces of the snakes.

By adjusting the coefficients $\alpha(s)$ and $\beta(s)$, the curve moves toward the object with proper elasticity and rigidity, i.e. continuity and smoothness. When the curve is closed to the edge of the object, the external energy takes effect. The external energy includes the image energy and constraints, i.e.

$$E_{ext}\left(\mathbf{v}(s)\right) = E_{img}\left(\mathbf{v}(s)\right) + E_{con}\left(\mathbf{v}(s)\right). \tag{6.3}$$

The first term of (6.3) is the image energy which depends on the information of the image. It attracts the curve towards the distinguish features of the image, such as lines and edges. The second term is the constraint term which is some form of high level image understanding, mostly imposed by users. In [31], the constraint term is manipulated by inserting a volcano icon which pushes the snake away. This is useful to avoid the curving convergent to an undesired local minimum. However, it is not easy and convenient for users. Therefore, the constraint term is not considered here. In such case, (6.1) can be rewritten as

$$J_{snake}(\mathbf{v}(s)) = \int_0^1 E_{int}\left(\mathbf{v}(s)\right) + E_{img}\left(u(\mathbf{v})\right) ds\,. \tag{6.4}$$

The image energy given by Kass is a weighted combination of three energy functionals, including lines, edges and terminations, as below

$$E_{img} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term}, \tag{6.5}$$

where w_{line}, w_{edge}, and w_{term} are weights as attached to each energy term. By adjusting these weights, the closed curve can be attracted to different features of the image. In the following study, the image energy is chosen as according to the features of an image itself. For instance, for an image with line bar only, E_{img} can be chosen as the intensity of the image[39], i.e.

$$E_{img}(x, y) = -u(x, y), \tag{6.6}$$

or its Gaussian blurring version

$$E_{img}(x, y) = -G_{\sigma}(x, y) * u(x, y), \tag{6.7}$$

which is used to make the boundary blurred for increasing the range of the capture of the active contour. However, such σ should not be too large to make edges delocalised. The edge detection by using (6.6) can be shown in Figure 6.2.

Figure 6.2: Active contour model with energy function (6.6).

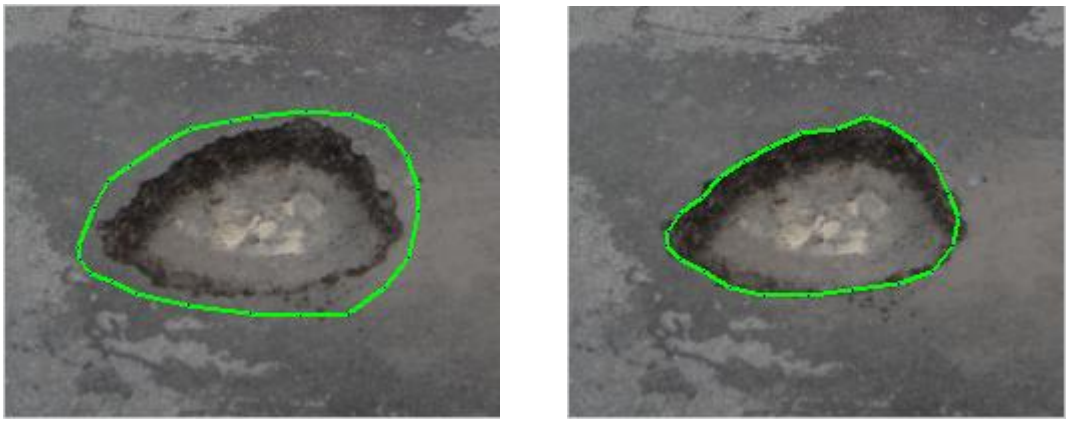
For grey-level images, typical image energy which leads a closed curve to the desired boundary can be designed as

$$E_{img}(x,y) = -\left|\nabla u(x,y)\right|^2, \tag{6.8}$$

or

$$E_{img}(x,y) = -\left|\nabla G_\sigma * u(x,y)\right|^2. \tag{6.9}$$

The minimisation of the integral of E_{img} is equivalent to attract the curve moving to the points at which the magnitude of gradient ∇u is big. This means that the curve stops slithering around edges of an object as observed in the Figure 6.3.



(a) the initial snake (b) the final snake

Figure 6.3: Active contour model using the energy (6.9).

6.1.2 Behavioural Analysis of Internal Energy

The effect of the internal energy is discussed in this section and the two coefficients α and β in (6.2) are investigated against their effects on the elasticity and curvature terms of the internal energy functional and the snake model.

The first term $\|\mathbf{v}'(s)\|^2$ in (6.2) is the first derivative of the snake, which measures the elasticity of the snake. It means that this term will have a high value where the curve is stretched. The second term $\|\mathbf{v}''(s)\|^2$ in (6.2) provides a curvature control, which attains a high value in that part of the snake where the curve has a sharp bent. The role of these two terms is examined below through the use of geometry.

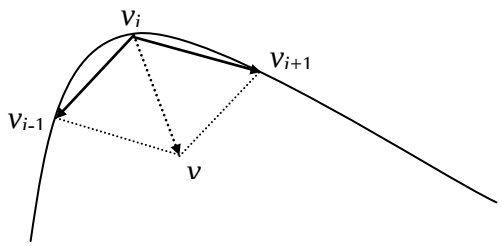


Figure 6.4: Geometry analysis

Considering three arbitrary adjacent vertices, v_{i-1}, v_i and v_{i+1},on the snake as depicted in Figure 6.4. Vectors $\overrightarrow{v_i v_{i-1}}$ and $\overrightarrow{v_i v_{i+1}}$ are

$$\overrightarrow{v_i v_{i-1}} = v_{i-1} - v_i, \tag{6.10}$$

and

$$\overrightarrow{v_i v_{i+1}} = v_{i+1} - v_i. \tag{6.11}$$

Note that the terms $\|\mathbf{v}'(s)\|^2$ and $\|\mathbf{v}''(s)\|^2$ can be expressed by the discretised forms,

$$\|\mathbf{v}'(s)\|^2 = (v_i - v_{i-1})^2, \tag{6.12}$$

and

$$\|\mathbf{v}''(s)\|^2 = (v_{i+1} + v_{i-1} - 2v_i)^2, \tag{6.13}$$

respectively. Furthermore, (6.13) can be rearranged as

$$\|\mathbf{v}''(s)\|^2 = \left((v_{i-1} - v_i) + (v_{i+1} - v_i)\right)^2. \tag{6.14}$$

Following (6.10), (6.11) and Figure 6.4, the two terms $\|\mathbf{v}'(s)\|^2$ and $\|\mathbf{v}''(s)\|^2$ can be represented respectively by the vector forms, i.e.

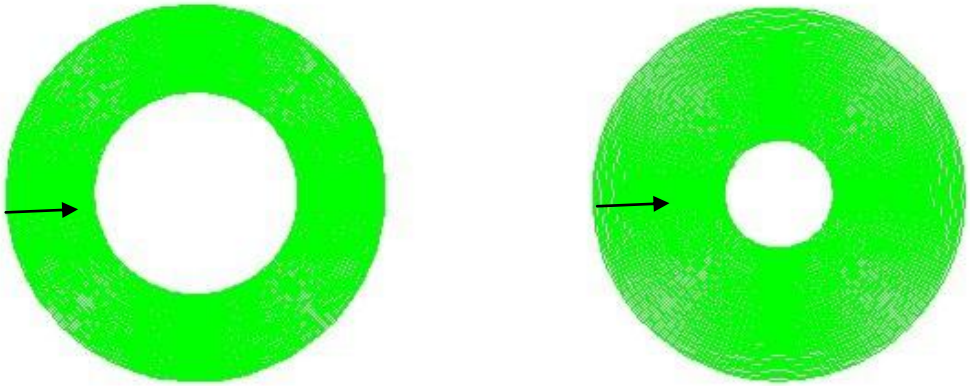
$$\|\mathbf{v}'(s)\|^2 = \left\|\overrightarrow{v_i v_{i-1}}\right\|^2, \tag{6.15}$$

and

$$\|\mathbf{v}''(s)\|^2 = \left((v_{i-1} - v_i) + (v_{i+1} - v_i)\right)^2 = \left(\overrightarrow{v_i v_{i-1}} + \overrightarrow{v_i v_{i+1}}\right)^2 = \left\|\overrightarrow{v_i v}\right\|^2. \tag{6.16}$$

Therefore, minimisation of the integral of the two terms $\|\mathbf{v}'(s)\|^2$ and $\|\mathbf{v}''(s)\|^2$ in (6.2) is equivalent to minimisation of the integral of the norms of vector $\overrightarrow{v_i v_{i-1}}$ and $\overrightarrow{v_i v}$ respectively. In other words, this optimal problem is to find a minimum distance from vertex v_{i-1} to vertex v_i, i.e. minimum length of the curve in continuous form, and a minimum norm of vector $\overrightarrow{v_i v}$ which can be obtained when vertices v_{i-1}, v_i and v_{i+1} are on a straight line. In the extreme case, this means the curvature of the curve is 0. From this point of view, the same conclusion is drawn as the previous analysis.

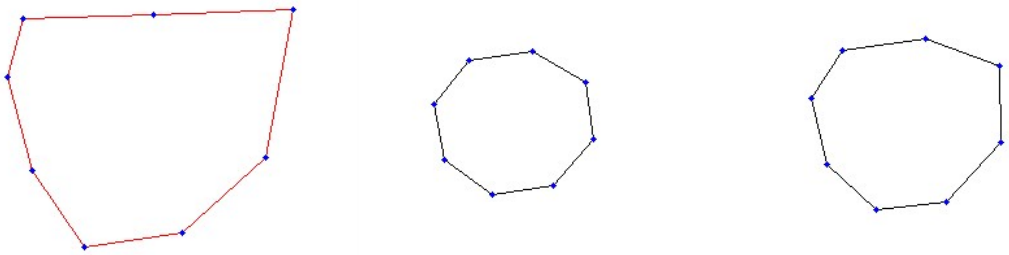
The influence of the elasticity term and curvature term is tremendously controlled by the two weighted coefficients α and β. Generally speaking, the coefficient α controls the rate of convergence of the curve. The larger the value α is, the higher the convergence rate. In Figure 6.5 below, the outer circles in (a) and (b) are the initial curves whereas the inner circles are the curves after 2000 iterations. It can be seen that under all other configurations being the same, the larger values of α leads to higher rate of convergence. Therefore, α is well known as the elastic coefficient.



(a) $\alpha = 0.5$ (b) $\alpha = 1$

Figure 6.5: Effect of α, in both cases $\beta = 0.05$, and the number of iterations is 2000.

Following the analysis above, since $\|\mathbf{v}''(s)\|^2$ is the norm of the curvature $|\kappa|$, such that minimisation of the integral of $\|\mathbf{v}''(s)\|^2$ requires that the curve to be as smooth as possible. The coefficient β is used to adjust the curve moving towards the object along the normal direction of the curve. When β is large, the curve would be stiff whereas the curve would be soft to bend when β is small. In order to make this clearer, two comparative demonstrations are designed as shown in Figure 6.6. The initial curve was designed with a sharp angle on the top right corner and after 5 iterations, it can be noted that that angle becomes bigger with larger value of β than that with smaller value of β. Therefore, when β is large, the curve tends to smoother such that the ability against bending is stronger than the case when β is small. Large values of β make the second order term larger than other terms such that the minimisation would occur when the curve is smoother approaching a circle for a closed curve and a straight line for an unclosed contour. Therefore, β is sometimes known as the stiffness coefficient.



(a) Initial snake (b) β=0.5 (c) β=0.01

Figure 6.6: Effect of β, α=0.5 and the number of iteration is 5.

Figure 6.5 and Figure 6.6 show that the internal energy is used to keep the curve as short and smooth as possible. However, the values of two coefficients have to be avoided being too great. Otherwise, overlarge values of α would make the curve shrink towards a point whereas overlarge values of β would make the curve too stiff to converge to a concavity. Both cases fail to detect the boundary of an object.

Based on these investigations, it can be seen that the elasticity and stiffness of active contour curve, i.e. continuity and smoothness, are well preserved by choosing suitable values of the coefficients, α and β. In some cases, since that the curve is becoming smoothing when minimising the length of the curve, the curvature term is omitted, as analysed in the geodesic active contour model (GAC) [40].

6.1.3 Numerical Implementation and Demonstrations

Using the result in section 2.2, the minimisation of the snake model

$$J_{snake}(\mathbf{v}(s)) = \int_0^1 \alpha(s)\left\|\mathbf{v}'(s)\right\|^2 + \beta(s)\left\|\mathbf{v}''(s)\right\|^2 + E_{img}\left(u(\mathbf{v})\right)ds, \tag{6.17}$$

requires the solutions of the Euler-Lagrange equations are

$$-\frac{\partial}{\partial s}\left\{\alpha(s)\frac{\partial x(s,t)}{\partial s}\right\} + \frac{\partial^2}{\partial s^2}\left\{\beta(s)\frac{\partial^2 x(s,t)}{\partial s^2}\right\} + \frac{\partial E_{img}}{\partial x} = 0, \tag{6.18}$$

and

$$-\frac{\partial}{\partial s}\left\{\alpha(s)\frac{\partial y(s,t)}{\partial s}\right\}+\frac{\partial^2}{\partial s^2}\left\{\beta(s)\frac{\partial^2 y(s,t)}{\partial s^2}\right\}+\frac{\partial E_{img}}{\partial y}=0\,. \tag{6.19}$$

The steepest decent method may be applied to solve the two PDES,

$$\frac{\partial x(s,t)}{\partial t}=-\frac{\partial}{\partial s}\left\{\alpha(s)\frac{\partial x(s,t)}{\partial s}\right\}+\frac{\partial^2}{\partial s^2}\left\{\beta(s)\frac{\partial^2 x(s,t)}{\partial s^2}\right\}+\frac{\partial E_{img}}{\partial x}, \tag{6.20}$$

and

$$\frac{\partial y(s,t)}{\partial t}=-\frac{\partial}{\partial s}\left\{\alpha(s)\frac{\partial y(s,t)}{\partial s}\right\}+\frac{\partial^2}{\partial s^2}\left\{\beta(s)\frac{\partial^2 y(s,t)}{\partial s^2}\right\}+\frac{\partial E_{img}}{\partial y}\,. \tag{6.21}$$

in order to solve the minimisation of the energy functional (6.17). The two equations can be rewritten in a vector form, that is,

$$\frac{\partial \mathbf{v}(s,t)}{\partial t}=-\frac{\partial}{\partial s}\left\{\alpha(s)\frac{\partial \mathbf{v}(s,t)}{\partial s}\right\}+\frac{\partial^2}{\partial s^2}\left\{\beta(s)\frac{\partial^2 \mathbf{v}(s,t)}{\partial s^2}\right\}+\nabla E_{img}(u(\mathbf{v}))\,. \tag{6.22}$$

If the initial and boundary conditions are considered, the evolving PDE of the snake model can be written in the form of

$$\begin{cases}\dfrac{\partial \mathbf{v}(s,t)}{\partial t}=-\dfrac{\partial}{\partial s}\left\{\alpha(s)\dfrac{\partial \mathbf{v}(s,t)}{\partial s}\right\}+\dfrac{\partial^2}{\partial s^2}\left\{\beta(s)\dfrac{\partial^2 \mathbf{v}(s,t)}{\partial s^2}\right\}+\nabla E_{img}(u(\mathbf{v}))\\ \mathbf{v}(s,0)=\mathbf{v}_0\\ \mathbf{v}(0,t)=\mathbf{v}(1,t)\\ \dfrac{\partial \mathbf{v}(s,t)}{\partial t}\Big|_{t=0}=0\end{cases}. \tag{6.23}$$

Here, the second equation in (6.23) is the initial curve and the third means that the curve is closed. The last equation indicates that the initial speed of the curve is zero. Based on these conditions, the behaviour of the snake is controlled by (6.23).

With (6.23), the next step is to provide a discrete formulation to approximate the solution. There are a few different methods to find the numerical solution of snake model. For example, finite difference method [31], finite element method [104], dynamic programming [105] and greedy method [106], etc. In this dissertation the finite difference method is applied to solve the snake model (6.23) which is described below.

The closed snake curve $\mathbf{v}(s)$ where s is continuous in the interval [0, 1] is discretised into a series of points $\mathbf{v}(s_i)$ $(i = 1, 2, \ldots n)$ with $\mathbf{v}(s_1) = \mathbf{v}(s_n)$. Suppose the time step is Δt and the curve is equally divided into a small distance denoted h. Therefore, the left-hand side of the first equation in (6.23) can be discretised as

$$\begin{aligned}
&\frac{\mathbf{v}^t(s_i) - \mathbf{v}^{t-1}(s_i)}{\Delta t} = \\
&-\frac{1}{h}\left\{\alpha(s_{i+1})\frac{\mathbf{v}^t(s_{i+1}) - \mathbf{v}^t(s_i)}{h} - \alpha(s_i)\frac{\mathbf{v}^t(s_i) - \mathbf{v}^t(s_{i-1})}{h}\right\} \\
&+\frac{1}{h^2}\left\{\beta(s_{i+1})\frac{\mathbf{v}^t(s_{i+2}) - 2\mathbf{v}^t(s_{i+1}) + \mathbf{v}^t(s_i)}{h^2} - 2\beta(s_i)\frac{\mathbf{v}^t(s_{i+1}) - 2\mathbf{v}^t(s_i) + \mathbf{v}^t(s_{i-1})}{h^2}\right. \\
&\left.+\beta(s_{i-1})\frac{\mathbf{v}^t(s_i) - 2\mathbf{v}^t(s_{i-1}) + \mathbf{v}^t(s_{i-2})}{h^2}\right\} + \nabla E_{img}(u(\mathbf{v}^{t-1}(s_i)))
\end{aligned} \tag{6.24}$$

The above equation can be rearranged according to the snake points $\mathbf{v}(s_i)$, that is,

$$\begin{aligned}
\frac{\mathbf{v}^t(s_i) - \mathbf{v}^{t-1}(s_i)}{\Delta t} &= \frac{\beta(s_{i-1})}{h^4}\mathbf{v}^t(s_{i-2}) + \left(-\frac{2\left(\beta(s_i) + \beta(s_{i-1})\right)}{h^4} - \frac{\alpha(s_i)}{h^2}\right)\mathbf{v}^t(s_{i-1}) \\
&+\left(\frac{\beta(s_{i+1}) + 4\beta(s_i) + \beta(s_{i-1})}{h^4} + \frac{\alpha(s_{i+1}) + \alpha(s_i)}{h^2}\right)\mathbf{v}^t(s_i) \\
&+\left(-\frac{2\left(\beta(s_{i+1}) + \beta(s_i)\right)}{h^4} - \frac{\alpha(s_{i+1})}{h^2}\right)\mathbf{v}^t(s_{i+1}) \\
&+\frac{\beta(s_{i+1})}{h^4}\mathbf{v}^t(s_{i+2}) + \nabla E_{img}(u(\mathbf{v}^{t-1}(s_i)))
\end{aligned} \tag{6.25}$$

Combining with the boundary condition, (6.25) holds for all $s_i, i = 1, 2, \ldots n$.

Let

$$
\begin{aligned}
a(s_i) &= \frac{\beta(s_{i-1})}{h^4}, \\
b(s_i) &= \left(-\frac{2\left(\beta(s_i)+\beta(s_{i-1})\right)}{h^4} - \frac{\alpha(s_i)}{h^2} \right), \\
c(s_i) &= \left(\frac{\beta(s_{i+1})+4\beta(s_i)+\beta(s_{i-1})}{h^4} + \frac{\alpha(s_{i+1})+\alpha(s_i)}{h^2} \right), \\
d(s_i) &= \left(-\frac{2\left(\beta(s_{i+1})+\beta(s_i)\right)}{h^4} - \frac{\alpha(s_{i+1})}{h^2} \right), \\
e(s_i) &= \frac{\beta(s_{i+1})}{h^4},
\end{aligned}
$$

and denote a new coefficient matrix as

$$
A = \begin{bmatrix}
c(s_1) & d(s_1) & e(s_1) & 0 & \cdots & a(s_1) & b(s_1) \\
b(s_2) & c(s_2) & d(s_2) & e(s_2) & 0 & \cdots & a(s_2) \\
a(s_3) & b(s_3) & c(s_3) & d(s_3) & a(s_3) & 0 & \cdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\vdots & & & & & & \\
e(s_{N-1}) & 0 & \cdots & a(s_{N-1}) & b(s_{N-1}) & c(s_{N-1}) & d(s_{N-1}) \\
d(s_N) & e(s_N) & 0 & \cdots & a(s_N) & b(s_N) & c(s_N)
\end{bmatrix}, \tag{6.26}
$$

In such a case, (6.25) can be rewritten in a matrix system form as

$$
\frac{\mathbf{v}^t(s_i) - \mathbf{v}^{t-1}(s_i)}{\Delta t} = A\mathbf{v}^t - \nabla E_{img}(u(\mathbf{v}^{t-1}(s_i))). \tag{6.27}
$$

Therefore, the numerical solution for the snake model can be addressed below，

$$
\mathbf{v}^t(s_i) = (I - \Delta t A)^{-1}(\mathbf{v}^{t-1}(s_i) - \nabla E_{img}(u(\mathbf{v}^{t-1}(s_i)))) \tag{6.28}
$$

Here, I is the identity matrix. By using iteration, the final snake can stop at the desired boundary of an object. Generally, the choice of time step is crucial to the speed of the curve evolution. Large values of the time step make the curve move faster. However, out of stability consideration, it is necessary to keep the time step small and avoid the curve stepping over the salient feature of the object.

To overview how the snake model works in object detection and the effect of the coefficients, a series of demonstrations are given below (The image size is 256×256). Figure 6.7 (a) is the

initial snake and the external image energy employed is (6.7). The parameters are listed in Table 6.1.

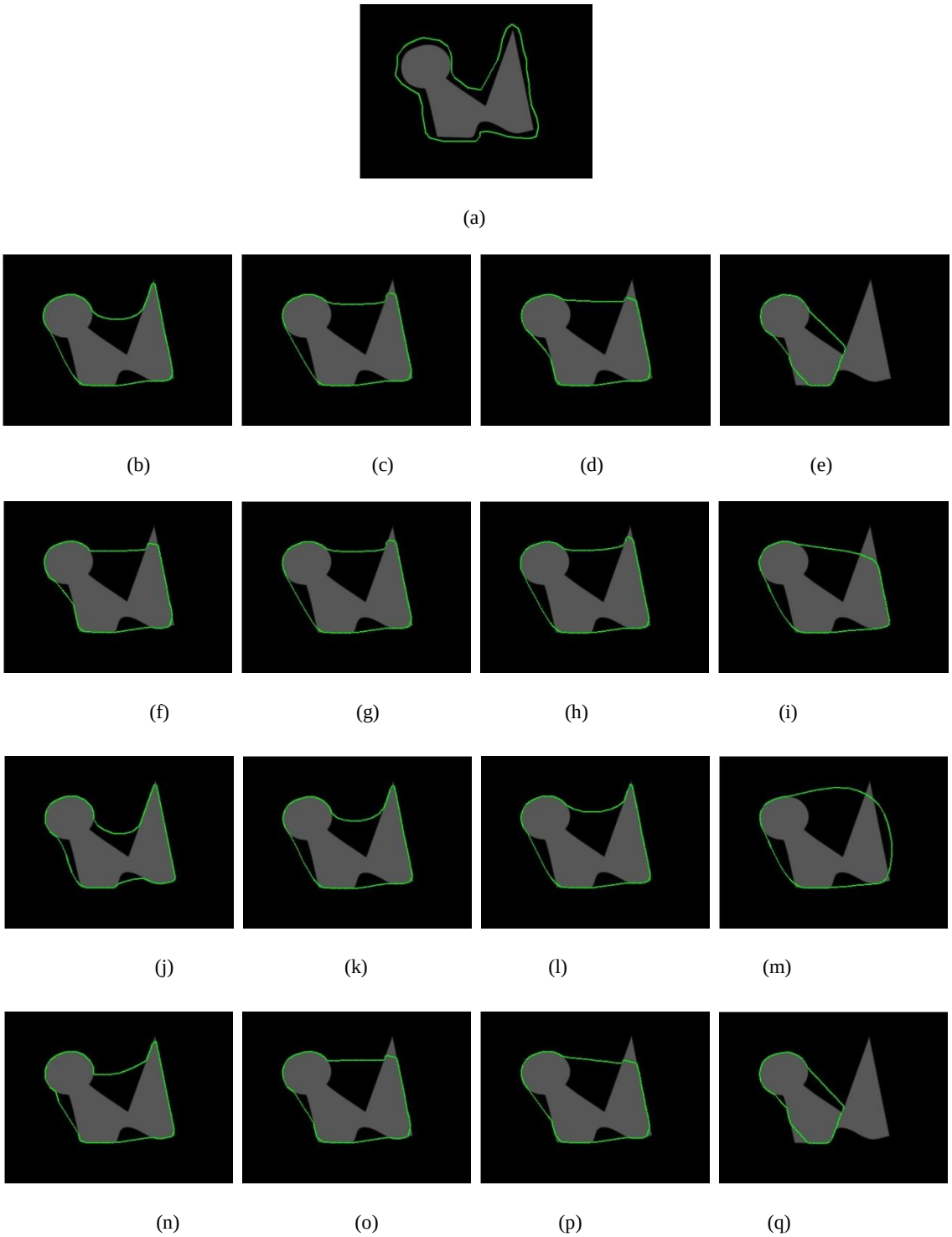


Figure 6.7: Demonstrations of snake model.

Paras / Figure index	α	β	iteration	Paras / Figure index	α	β	iteration
(b)	0.0005	0.005	6000	(j)	0	0.0005	6000
(c)	0.005	0.005	6000	(k)	0	0.005	6000
(d)	0.01	0.005	6000	(l)	0	0.01	6000
(e)	0.05	0.005	6000	(m)	0	0.1	6000
(f)	0.005	0.0005	6000	(n)	0.0005	0	10000
(g)	0.005	0.005	6000	(o)	0.005	0	10000
(h)	0.005	0.01	6000	(p)	0.01	0	10000
(i)	0.005	0.05	6000	(q)	0.05	0	10000

Table 6.1: Coefficient configuration for Figure 6.7.

The demonstrations show that when the initial snake is put closed to the boundary of the object, it would move towards the desired boundary. However, when there are cusps or concavity, the curve cannot converge to it. These can be observed from all results. The other phenomenon can be observed from the demonstrations is the influence of the coefficients. From Figure 6.7 (b)-(e) and (n)-(q), one can see that when the value of β keeps unchanged, with the value of α increasing, the curve tends to be piecewise straight (d), (p) and overlarge values of α would make the curve step over the boundary (e), (q). From Figure 6.7 (f)-(i), one can observed that the curve keeps smoother as the value of β goes up (f), (g) and when when the value of β becomes bigger enough, the influence of stiffness term weights over those of the elasticity term and the energy term such that it makes the curve too smooth to stick to the boundary. This also can be more easily found in (m). From all figures, it can be seen that Figure 6.7(j) is the best result however it takes more time because without the elasticity term, the curve would slither slowly towards an object.

6.1.4 Summary

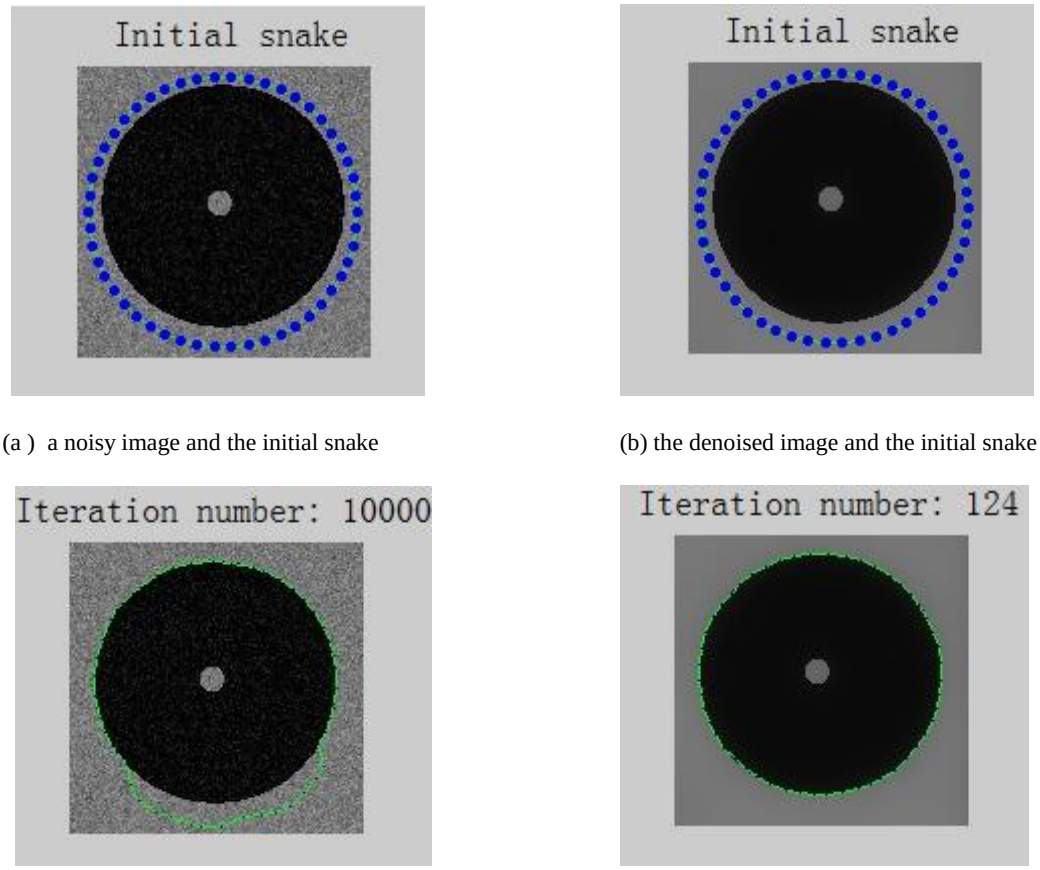
The active contour model (snake model) provides a determination procedure to use a geometric curve to detect an image contour. The idea is intuitive and easy to introduce prior knowledge (i.e., image information) into the model. As the first work which employs curve evolution for image segmentation, it highlights a new area and now there are numerous works established based on this idea. However, there are some intrinsic problems for this model. Firstly, the location of the initial curve affects the segmented result tremendously. Only curves being put close to the boundary of an object can provide the desired result. Secondly,

selections of the coefficients and the number of control points also have a huge influence on the final result. Thirdly, it cannot follow topological changes of objects which mean it can only detect one object each time. The last but not the least is that it is quite sensitive to the noise which can be addressed in the next section.

6.2 ACM Coupling with the Anisotropic Diffusion Model

6.2.1 Motivation

In Chapter 3, basic concepts of image noise and the reason of removing noise from an image are introduced and described. One example that the existence of noise makes the edge detection failed is shown there. Apart from this, the existence of noise also makes the active contour model low in performance, which can be depicted in the Figure 6.8.



(a) a noisy image and the initial snake

(b) the denoised image and the initial snake

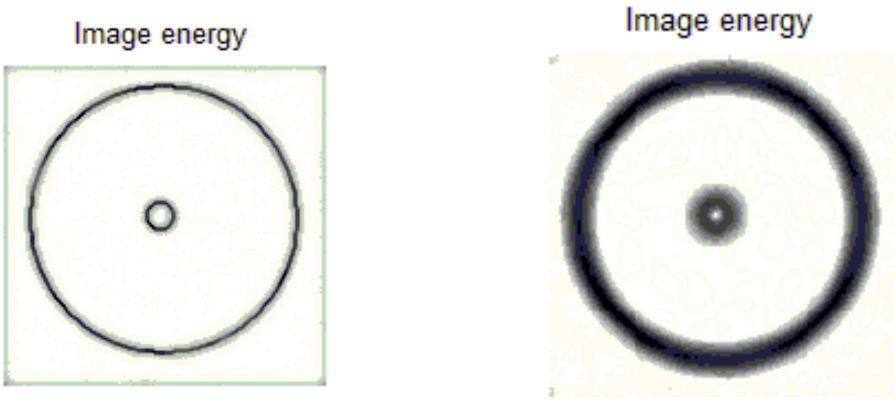
(c) the snake evolution of (a)

(d) the snake evolution of (b)

Figure 6.8: Influence of noise for the active contour model.

From Figure 6.8, it is illustrated that the snake takes more than 10000 iterations to converge to the circle where noise exists whereas only 124 iterations for the denoised circle, under the same circumstance (The snake itself, position, sampling, etc.). Therefore, snake in (d) evolves

much faster than that in (c), which means that the existence of noise slows down the convergent speed quite enormously. One possible solution is to use larger spacing step, however this may make the curve step over the boundary especially when the image energy is small. The other option is to use Gaussian convolution to the image before the segmentation procedure. With this, not only does it resist the noise from the image, but also extends the capture ability of the snake (Figure 6.9).



(a) $E_{img}(x,y) = -|\nabla u(x,y)|$ (b) $E_{img}(x,y) = -\left|\nabla\left(G_\sigma(x,y) * I(x,y)\right)\right|$

Figure 6.9: Image energy with and without Gaussian convolution.

Although the Gaussian convolution can expand the capture area, it may bring in fake boundary, such that the curve may not be stopped at the real boundary. Considering that the anisotropic diffusion model (P-M model) introduced in Chapter 4, on one hand, this model can smooth noise effectively. On the other hand, it can enhance the edge during backward diffusion, which is beneficial for stopping the curve at the boundary of an object. Based on this concept, this model can be introduced into the active contour model to build a noise-resistant model which is described in the next section.

6.2.2 Modelling

Based on the consideration made above and the fact that the active contour model is affected by the noise hugely, the improved framework for a noise-resistant active contour model is described by energy functional, that is,

$$E_{snake}(v(s),u) = \int_0^1 \alpha(s)\left\|\mathbf{v}'(s)\right\|^2 + \beta(s)\left\|\mathbf{v}''(s)\right\|^2 + E_{img}\left(\mathbf{v}(s)\right)ds + \lambda\int_D f\left(\left\|\nabla^2 u\right\|\right)dD\,. \quad (6.29)$$

Here, D means the area inside the evolving curve and the function $f(\cdot)$ can be chosen according to different requirements. The weighted coefficient $\lambda(v)$ is defined as

$$\lambda = \begin{cases} 1 & \text{if the pixel is closed to } v \\ 0 & \text{if the pixel is far away from } v \end{cases}, \tag{6.30}$$

which is used to avoid diffusion process in areas far away from the curve. In such case, the anisotropic diffusion filtering provides an iteratively enhanced edge map for the image energy. This can facilitate the segmentation of real boundaries of an object and sharpen edges with low gradients.

By using the variational method and steepest decent method, (6.29) can be solved by coupled PDEs as below,

$$\begin{cases} u_t = \lambda \nabla \cdot \left(\dfrac{f'(\|\nabla u\|)}{\|\nabla u\|} \nabla u \right) \\ \dfrac{\partial \mathbf{v}(s,t)}{\partial t} = -\dfrac{\partial}{\partial s}\left\{ \alpha(s) \dfrac{\partial \mathbf{v}(s,t)}{\partial s} \right\} + \dfrac{\partial^2}{\partial s^2}\left\{ \beta(s) \dfrac{\partial^2 \mathbf{v}(s,t)}{\partial s^2} \right\} + \nabla E_{img}(u_t(\mathbf{v})) \end{cases}. \tag{6.31}$$

Comparing with the P-M model in Chapter 4, it is easy to find that

$$c(s) = \frac{f'(s)}{s} \tag{6.32}$$

Figure 6.10 shows the noise-resistance functionality of this model. The coefficient is chosen as (4.14). The initial conditions are the same as those in Figure 6.8. It can shorten the convergent time efficiently. As for the computational cost, as it is well known that the P-M model is a very effective algorithm and the diffusion progress is restricted in a narrow band, comparing with the convergent time it saves, it is reasonable to introduce the new term into the active contour model.

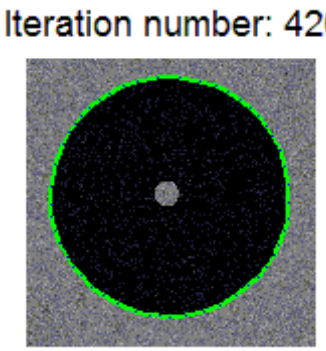
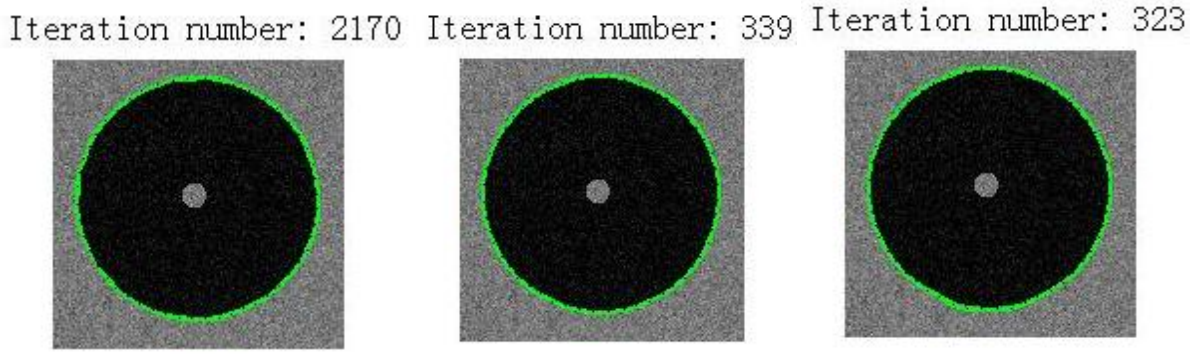


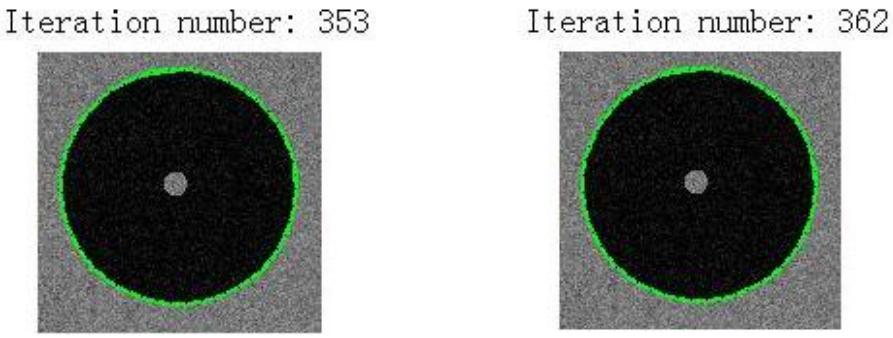
Figure 6.10: Noise-resistance active contour model.

6.2.3 Discussion

It is well known that the choice of coefficient in the P-M model is very important for the denoised result as discussed in the chapter 4. The coefficient determines that the model performs backward or forward diffusion. In order to show the effect of the diffusive behaviour on the convergent of the segmentation, apart from the coefficient used in section 6.2.2, other coefficients, including the two in the IPM and WYK model and the one in [14] are adopted below.



(a) Original ACM model (b) $c(s)=\frac{1}{1+(s/K)^2}$, $K=10$ (c) $c(s)=e^{-(s/K)^2}$, $K=10$



(d) $c(s) = s^{p-2},\ p = 1.3$ (e) $c(s) = \dfrac{1}{\sqrt{1+(s/K)^2}},\ K = 10$

Figure 6.11: Segmentation results with different coefficient in the ACM coupling with the P-M model

It can be seen that it takes much fewer number of iterations by using the ACM model coupling with the P-M model. The coefficient used in Figure 6.11(b) and (c) are the two coefficients suggested in the P-M model, which make the model diffuse backward when $s > K$ and forward when $s < K$. The other two coefficients is the one proposed in the IPM model and the one used in the WYK model, which always make the model diffuse forward. Due to the backward diffusion sharpening edges, the coefficients given in the P-M model perform slightly better than the other two. Therefore, the ACM model coupling with the anisotropic diffusion model can detect the desired objects under the existence of noise.

6.3 Closure

In this chapter, the active contour model (ACM or Snake) was carefully examined in detail. It explained the reason that the curve move towards the desired object and discussed the internal energy in the model. Furthermore, a geometric description of the internal energy was provided to understand the behaviours of the ACM model. At the end of this chapter, the ACM model coupling with the anisotropic diffusion model which was discussed in the previous chapters were proposed and the experimental result shows that such coupling can build up a noise-resistance segmentation model.

Chapter 7. EXPERIMENTAL DEMONSTRATION

In this chapter, a series of experiments related to image denoising are designed to test both of the existing models and the proposed improved models in this thesis. The segmentation experiments are provided in Chapter 6. The computing environment consisted of a PC workstation with Intel Core Duo E8500 running Windows 7 Enterprise, 64-bit Operating System, supported by Visual C++6.0 and Matlab 2010a. In addition, the 2-D denoising programs use C++ programming language without using independent libraries and 1-D denoising programs and segmentation programs use Matlab for implementation. There in one code example provided in Appendix.

7.1 Second Order PDE Image Denoising

In this section, several methods including one famous linear filter (Gaussian filter), one classical nonlinear filter (median filter), the anisotropic diffusion model (P-M model) and the modified second order PDE method in section 5.1 (IPM model) are used in the experiments for 1-D signals and 2-D images.

7.1.1 1-D Signal Denoising

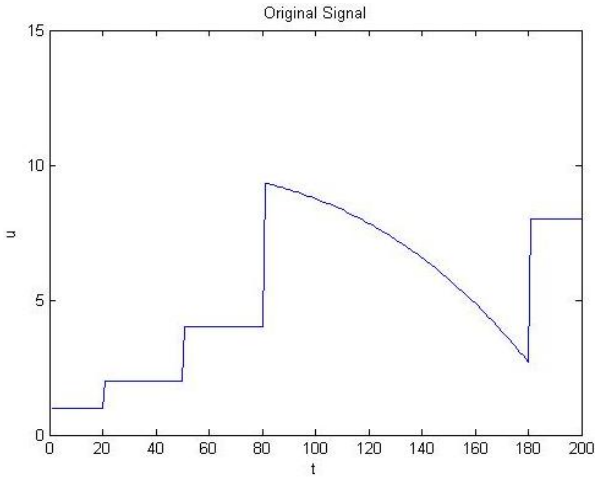
Different models as mentioned above were applied to two 1-D signals with different Gaussian noise. The Gaussian noise applied to the signals is shown in Table 7.1. The results are displayed in Figure 7.1 for the two signals and the related parameters are shown in the Table 7.2. The original signal (depicted as a black curve) is plotted on the denoised signals for comparison.

t / *Gaussian variance* / *Signal*	(0,20]	(20,50]	(50,100]	(100,150]	(150,200]
a	0.1	0	0.2	0.3	0
b	0	0	1	5	0

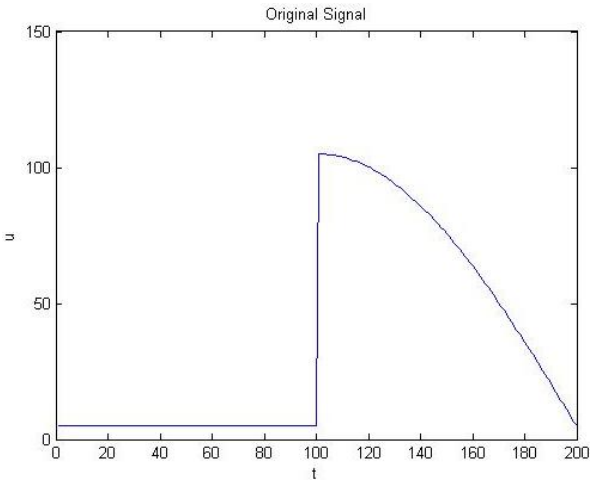
Table 7.1: Configuration of signals

Paras / *Value* / *Model*	Window size	σ	*K*	*p*	iteration	time step
Median Filter	3	N/A	N/A	N/A	N/A	N/A
Gaussian Filter	15	5	N/A	N/A	N/A	N/A
P-M Model	N/A	N/A	10	N/A	15	0.5
IPM Model	N/A	N/A	N/A	1.5	15	0.125

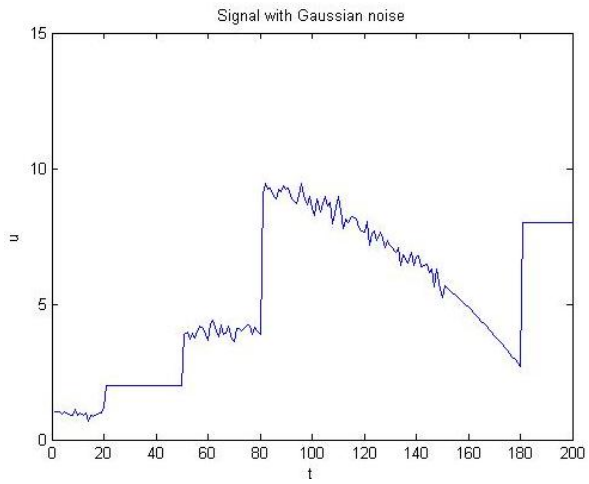
Table 7.2: Model parameter configuration for both signals.



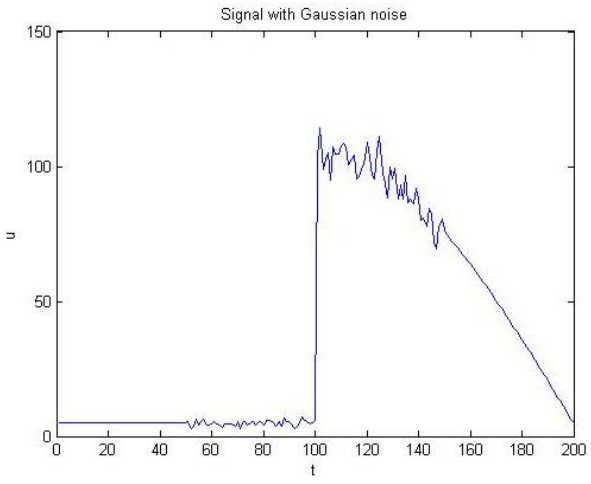
(a) signal *a*



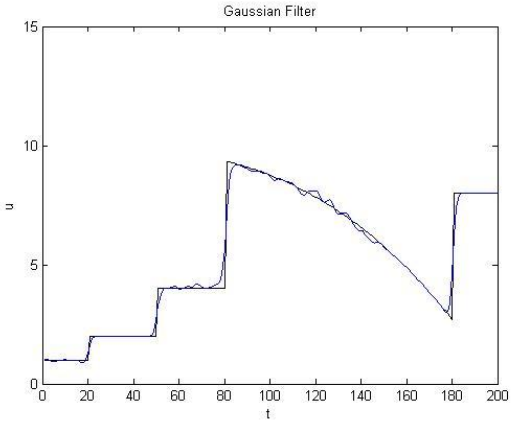
(b) signal *b*



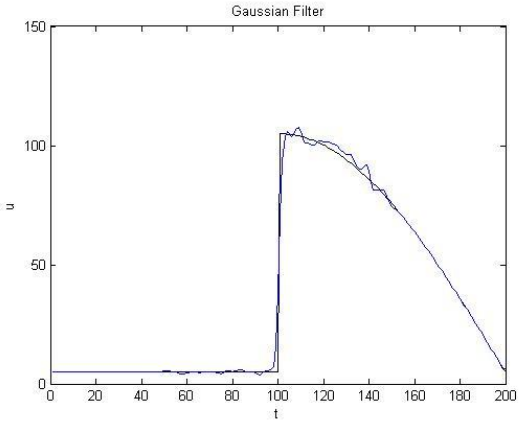
(c) noisy signal *a*



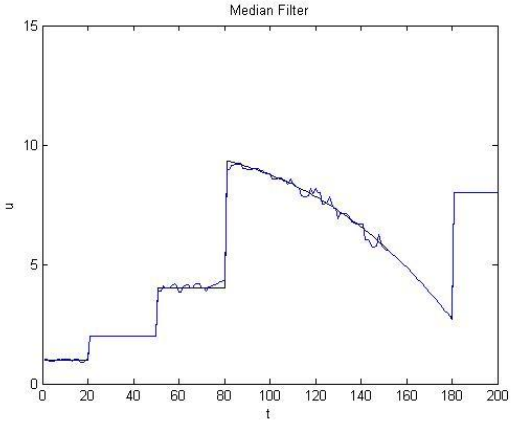
(d) noisy signal *b*



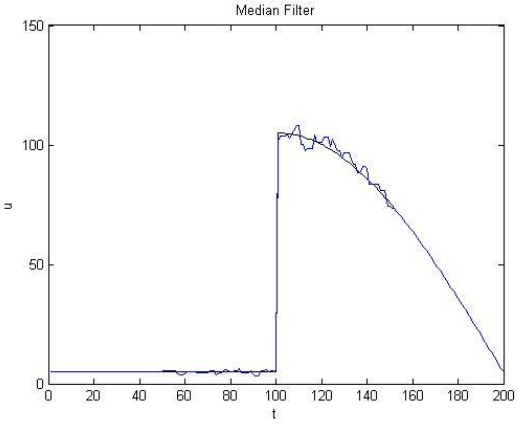
(e) denoised signal *a* by Gaussian filter



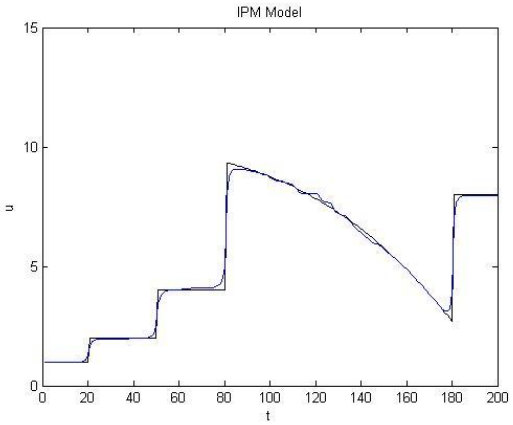
(f) denoised signal *b* by Gaussian filter



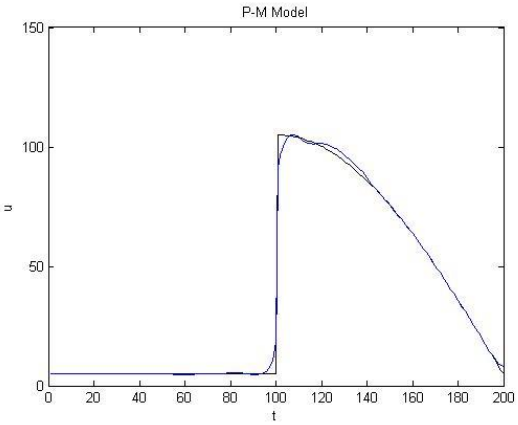
(g) denoised signal *a* by Median filter



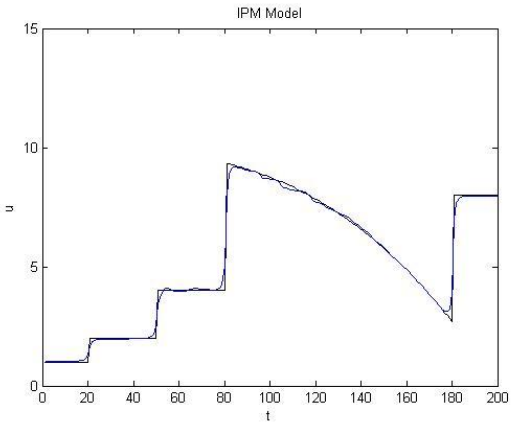
(h) denoised signal *b* by Median filter



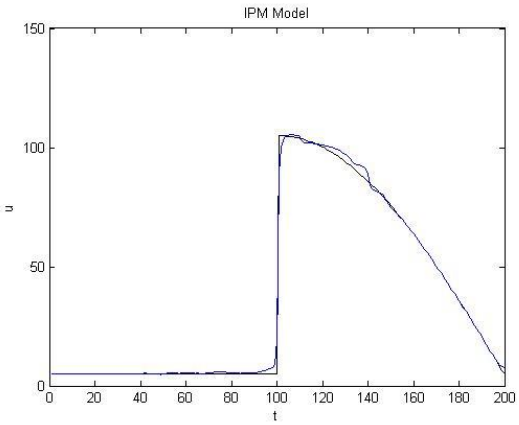
(i) denoised signal *a* by P-M model



(j) denoised signal *b* by P-M model



(k) denoised signal *a* by IPM model



(l) denoised signal *b* by IPM model

Figure 7.1: 1-D signal denoising by using second order PDE models and filtering methods.

Figure 7.1 (a) and (b) are two original signals and Figure 7.1 (c) and (d) are the corresponding signals after applying different Gaussian noise to Figure 7.1 (a) and (b) respectively. From figures in both columns, it can be seen that the Gaussian filter and median filter can only remove a certain amount of noise from the signals in both cases. Signals without noise were preserved better than the P-M and IPM models. For the purpose of removing noise from the noisy images, these two methods did not perform well. From Figure 7.1 (i), (j) and (k), (l), it can be observed that both P-M model and IPM model remove the noise from the two signals and provide smooth signals for future processing. However, by more careful comparison between two groups of results, it is not difficult to see that the IPM model can preserve edges better than the P-M model (especially in the area $t \in [80,100]$ and other edge areas from Figure 7.1 (i) and (k)), though both models would lead to loss of edges to some extent. In order to compare the similarity between the denoised signals and the original signal quantitatively, the L_1-norm defined as below is employed in this thesis.

$$Error = \int |f(x) - g(x)| dx \tag{7.1}$$

where $f(x)$ and $g(x)$ means the original and denoised signal. The results of Figure 7.1 are shown in Table 7.3 as below.

Denoised signal / *Error* / *Model*	*a*	*b*
Noisy signal	21.64	243.15
Median Filter	14.83	160.81
Gaussian Filter	22.42	204.60
P-M Model	21.57	129.58
IPM Model	17.92	126.93

Table 7.3: Error comparison of Figure 7.1

The error of signal *a* after using a median filter is the smallest as shown in Table 7.3. The reason is that the amount of noise in signal *a* is relatively small so that whenever the median filter is employed, it keeps most areas unchanged because signal *a* shows piecewise properties. However, when the amount of noise is large, it does not achieve the smallest error. Therefore, from the description above, the IPM model shows better result than the other models both in visualisation and in error comparison.

7.1.2 2-D Image Denoising

In this section, the same models mentioned in 7.1.1 and the TV model proposed in [19] are tested on images. For simplicity, two benchmarking images, Lena and Camera, which are noise-free images, are used in this and following sections, which are shown in Figure 7.2 (a) and (b). The corresponding noisy images with 10dB Gaussian noise are given in Figure 7.2 (c) and (d). Two groups of demonstration are designed to compare the advantages of each model. In order to inspect the tiny difference between different models, zoom-in images for Lena are given in the first experiment.

(a) Lena

(b) Camera

(c) noisy Lena

(d) noisy Camera

Figure 7.2: Benchmarking images and corresponding noisy images.

The first demonstration is to remove the noise from Figure 7.2 (c) by using the methods mentioned above. Details of configuration of each method are shown in Table 7.4 and the denoised results are provided in Figure 7.3 with some highlighted parts for comparison purpose.

Paras / *Value* / *Model*	Window size	σ	K	p	λ	iteration	time step
Median Filter	3	N/A	N/A	N/A	N/A	N/A	N/A
Gaussian Filter	5	3	N/A	N/A	N/A	N/A	N/A
P-M Model	N/A	N/A	10	N/A	N/A	15	0.25
TV model	N/A	N/A	N/A	N/A	0.25	20	N/A
IPM Model	N/A	N/A	N/A	1.3	N/A	15	0.25

Table 7.4: Configurations of Figure 7.3

(a) result by median filter

(b) partial zoom-in of (a)

(c) result by Gaussian filter

(d) partial zoom-in of (a)

(e) result by P-M model

(f) partial zoom-in of (c)

(g) result by TV model

(h) partial zoom-in of (g)

(i) result by IPM model

(j) partial zoom-in of (i)

Figure 7.3: Demonstration of image denoising of Lena by second order PDE models and typical methods.

From this demonstration, it can be seen that all of the methods presented can remove the Gaussian noise from the noisy image. However, by meticulous comparison with zoomed-in images, differences among various methods are considerable. The Gaussian filter cannot only remove noise from the noisy image, but also other details including edges. In such case, the whole image is blurred as shown in the results. The median filter, in another way, leads to a non-natural outlook of the result image and leaves the image with spurious blots. The last four images give the denoised results by the second order PDE models. Those show that both models can smooth the image and keep edges. The P-M model would sharpen edges however it fails to keep details especially in parts of the hair. From the zoomed-in images, one can also see that the result obtained by the IPM model has more smoothing edges without oscillation which is shown in Figure 7.3 (f). The “block effect” can be observed in both images obtained

by the P-M model and the IPM model but it is not severe due to the small number of iterations. However, the result obtained by the IPM model seems less severe and more natural than that provided by the P-M model. With very direct comparison, the conclusion can be drawn by subjective observation, that is, the IPM model has a better performance in image denoising that the P-M model and other classical and sophisticated methods. In order to give further persuasion, an objective comparison is given below, that is, PSNR, the definition of which is introduced in the last section of this chapter.

Model	PSNR
Noisy image	28.116
Median filter	30.198
Gaussian filter	26.610
P-M model	30.267
TV model	31.087
IPM model	30.912

Table 7.5: Objective measurement-PSNR for Figure 7.3

Table 7.5 obviously shows that the TV model provides the best PSNR result amongst these methods tested above. However, by comparing the zoom-in results from Figure 7.3(h) and (j), it shows that the IPM model provides more natural visualisation than that of TV model. Combined with the subjective analysis, the conclusion that the IPM model performs well in image denoising can be drawn. It not only smoothes noise in the image, but also keeps more details than the P-M model. In order to test the robustness of the improved model, another denoising demonstration applied on the image Camera.bmp is carried out. Configurations and Parameters are the same as those in Table 7.4, therefore, they are not displayed here again. Results are shown in Figure 7.4 and the PSNR comparison in Table 7.5.

(a) noisy image

(b) result by median filter

(c) result by Gaussian filter

(d) result by P-M model

(e) result by TV model

(f) result by IPM model

Figure 7.4: Demonstration of image denoising of Camera by second order PDE models and typical methods.

Models	PSNR
Noisy image	28.297
Median filter	26.363
Gaussian filter	22.625
P-M model	29.990
TV model	25.325
IPM model	30.142

Table 7.6: Objective measurement-PSNR for Figure 7.4

From Figure 7.4 and Table 7.6 it is not difficult to obtain the same conclusions as those drawn from the denoising demonstration on Lena. It can be seen that the PSNR of TV model is smaller by using the same configurations as those in Lena experiment. Therefore, it shows that the IPM model is more robust than the TV model. Furthermore, from the PSNR comparison in both demonstrations, the PSNR of denoising results by Gaussian filter and Median filter in Table 7.6 are evidently smaller than the noisy image (Camera) whereas the same phenomenon does not appear in Table 7.5. One possible reason for this is that the choice of parameters is not suitable for the image camera which leads to an undesirable result. However, from the view of visual effect, the noise in the image is almost removed by these two algorithms although not perfectly. Therefore, there may have some problems of using PSNR as the measurement for denoising quality, which is addressed and discussed in the last section of this chapter.

7.2 Fourth Order PDE image denoising

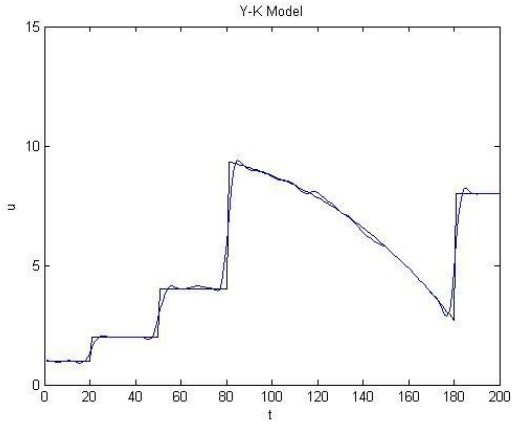
In this section, experiments by using fourth order PDE models are applied to image denoising. These models include the fourth order PDE model provided by Yu and Kaveh (Y-K model) and the two improved fourth order PDE models based on the Y-K model, that is, WYK model and AYK model, respectively. 1-D signals and 2-D images are both tested to show the denoising effect of each model.

7.2.1 1-D signal Denoising

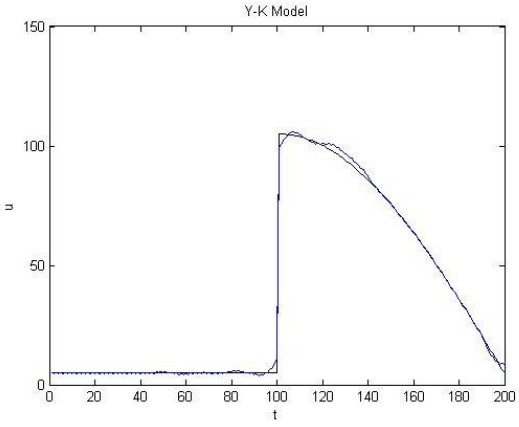
In the 1-D signal denoising demonstration for the fourth order PDE models, the same signals as those in Figure 7.1 are used here. Table 7.7 gives the configuration for each model and Figure 7.5 provides the denoising signals.

Paras / Value / Model	*K*	iteration	time step
Y-K model	0.5	2000	0.25
WYK model	0.3	400	0.25
AYK model	0.5	300	0.25

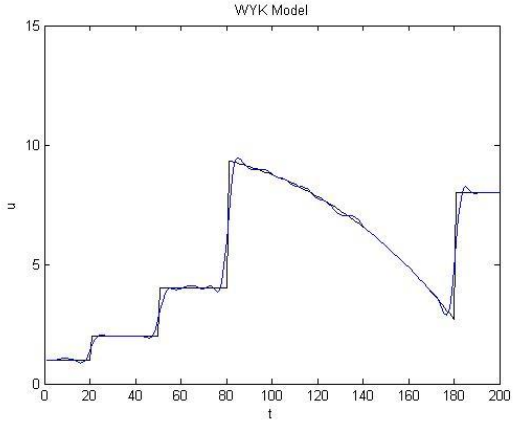
Table 7.7: Models' parameter configuration for signals in Figure 7.1 (c) and (d)



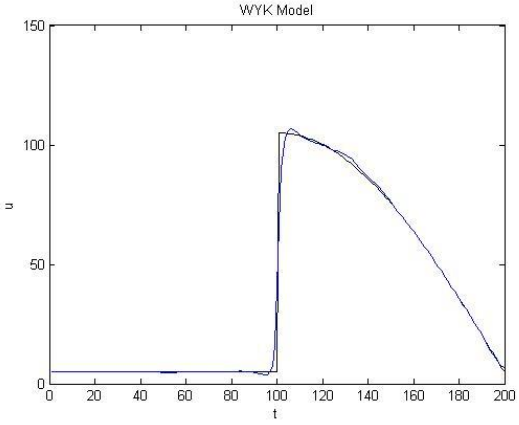
(a) denoised signal *a* by Y-K model



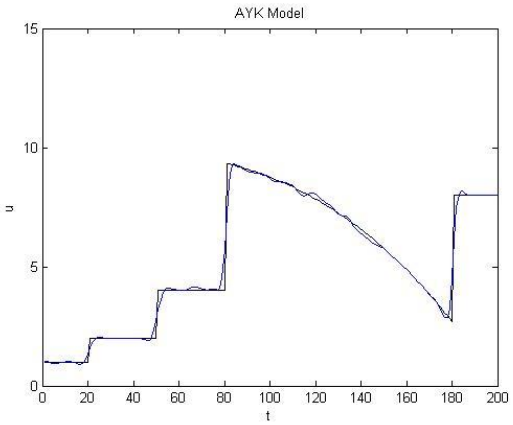
(b) denoised signal *b* by Y-K model

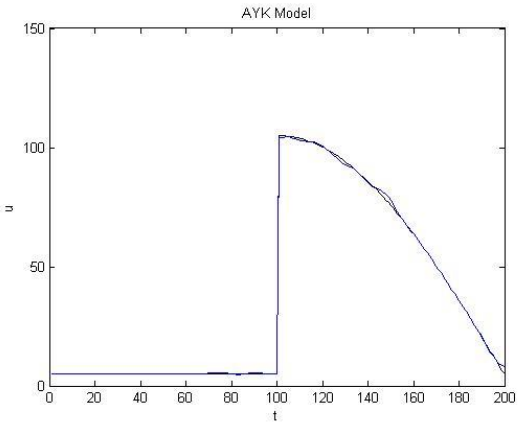


(c) denoised signal *a* by WYK model



(d) denoised signal *b* by WYK model





(e) denoised signal *a* by AYK model (f) denoised signal *b* by AYK model

Figure 7.5: 1-D signal denoising demonstration by fourth order PDE models.

From this demonstration, it can be seen that all three fourth order PDE models can remove the noise from the signal, although some details are lost, more or less. By comparing the results given by the Y-K and the two proposed models, one can observe that, for signal *a*, the performance of these three models are almost the same. For signal *b*, the AYK model provides the best result, whether in the flat area ($t \in [0,50]$, small oscillation in the Y-K model) or in the noisy area ($t \in [51,100]$, small oscillation in the Y-K model and WYK model). Same conclusion can be obtained by using (7.1) to calculate errors between the denoised signals and the original signal as shown below.

Denoised signal / *Error* / *Model*	*a*	*b*
Noisy signal	21.64	243.15
Y-K model	24.59	140.44
WYK model	23.22	134.35
AYK model	23.61	97.15

Table 7.8: Error comparison of Figure 7.5

7.2.2 2-D Image Denoising

In this section, the Y-K model, WYK model and AYK model are used to test their validity for removing noise from images. The two benchmarking images and their noisy images given by Figure 7.2 are used here together for the fourth order PDE demonstration. Configurations of each model are given in Table 7.9 and denoised images are provided in Figure 7.6.

Algorithms	Y-K model	WYK model	AYK model
	Figure(a) and (b)	Figure(c) and (d)	Figure(e) and (f)
Parameters	$K=10$ $\Delta t=0.25$ 500 iterations	$K=1$ $\Delta t=0.25$ 300 iterations	$K=10$ $\Delta t=0.25$ N/A

Table 7.9: Configurations of fourth order PDE models

(a) denoised Lena by Y-K model

(b) denoised Camera by Y-K model

(c) denoised Lena by WYK model

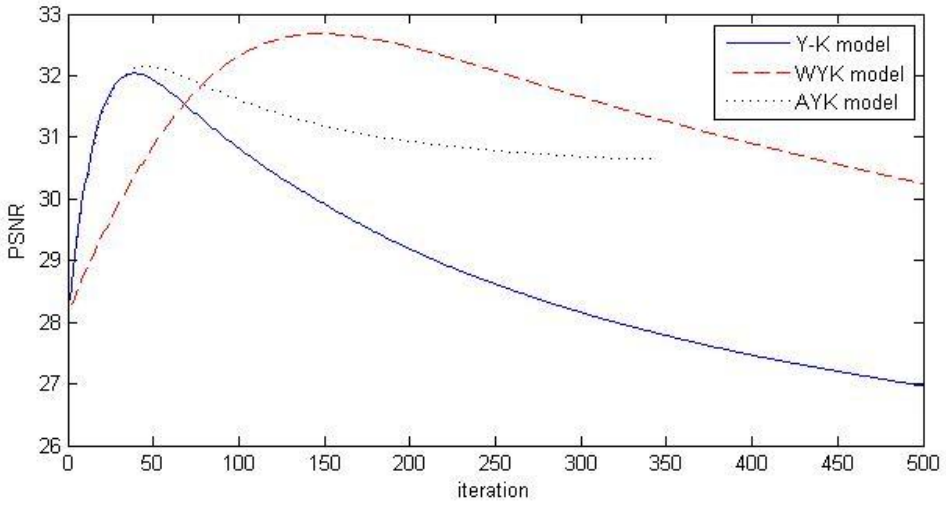
(d) denoised Camera by WYK model

(e) denoised Lena by AYK model

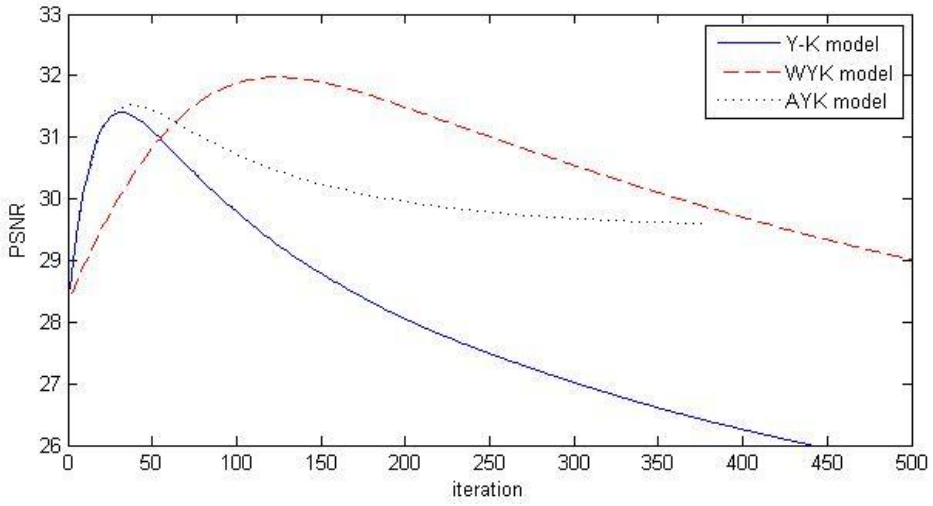
(f) denoised Camera by AYK model

Figure 7.6: Demonstration of image denoising of Camera by second order PDE models and typical methods.

Figure 7.6 shows the three models applied to the benchmark images, Lena and Camera. From the results given by the Y-K model, one can see that although noise is removed, some isolated speckles are brought in. The WYK model, on one hand, performs noise removal and keeps more details than the Y-K model. On the other hand, it still leads to isolated speckles. However, it relieves this symptom to some extent. The AYK model not only removes noise from the image, but also avoids isolated speckles successfully. Comparing with the results given by the WYK model, it leads to more details lost. Figure 7.7 provides the PSNR comparison amongst three models, which shows that the WYK and AYK models present better PSNR values than Y-K model and in the long term, the AYK model degrades the image more slowly than the WYK and Y-K model.



(a) Lena



(b) Camera

Figure 7.7: PSNR-Iteration Profile of the fourth order PDE models.

7.2.3 PSNR discussion

In this thesis, the only objective error measurement of image quality used is PSNR. In the area of image processing, there are some frequent metrics used for measure the quality of an image, such as MSE (mean-square-error), MAE (mean-absolute-error) and PSNR (peak-signal-noise-ratio). Their formulas can be expressed as below.

Given that the dimension of an image is $M \times N$, then

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[I(i, j) - u(i, j) \right]^2 \tag{7.2}$$

$$MAE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| I(i, j) - u(i, j) \right| \tag{7.3}$$

$$PSNR = \frac{255 \times 255}{\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[I(i, j) - u(i, j) \right]^2} \tag{7.4}$$

where $I(i, j), u(i, j)$ represents the original and the restored images pixels, respectively.

From these formulas, it is not difficult to find out that essentially, they are almost the same. Greater values of MSE or MAE (which means lower quality of a restored image) are corresponding to smaller values of PSNR. However, an image consists of pixels. These metrics presents the overall error between the restored image and the original image. Therefore, it does not absolutely mean that greater value of PSNR gives the better restored image. In order to show this idea, two restored images by the WYK model in the 2-D demonstration are displayed below for comparison. It can be seen that, although Figure 7.8 (a) obtains a greater value of PSNR, the noise is not totally removed from the image.

(a) iteration = 130, PSNR = 31.97

(b) iteration = 300, PSNR = 30.55

Figure 7.8: PSNR and image quality.

7.3 Closure

This chapter provides the numerical experiments for the denoising algorithms involved in this thesis. From the experiments, the conclusion can be drawn that the improved methods (IPM, WYK and AYK models) perform better denoising behaviour than the initial models (P-M, Y-K models). Issues regards to the error measurement of image quality also be discussed and further work will be focused in the future.

Chapter 8. CONCLUSIONS AND FUTURE WORK

This thesis presents the author's PhD work on the study of the PDE and variational methods in image processing, mostly in image denoising and segmentation. During this study, several contributions were made to improve the application of PDE on image processing.

8.1 Summary

The first contribution is to address and summarise three different methods from which to develop a PDE for image processing. These three ways show that a PDE can be derived from scale space theory, Gaussian convolution, and filtering respectively. On one hand, this strengthens the relation between PDE and other methods in image processing. On the other hand, it provides the validity of using PDE methods for image processing.

The second contribution of this thesis is to propose a new coefficient for the P-M model after analysing its behaviour and the reason of "block effect". The idea is based on the consideration that the P-M model is an ill-posed equation and out of this, if one needs to obtain a well-posed P-M model, the coefficient should satisfy the inequality $\phi'(s) \geq 0 \; for \; s \geq 0$, where $\phi(s) = sc(s)$ .Therefore, the coefficient $c(s) = s^{p-2}, 1 < p < +\infty$ is proposed to arrive at a well-posed P-M model. Experimental results showed that this coefficient can avoid backward diffusion such that it can weaken the "block effect" which is well known in the P-M model.

The third contribution is to discuss and explore the choice of *K* in the Y-K model and possible way of choosing the value of *K* was addressed. Although there is no adaptive value of *K* proposed, it still restricts the value of *K*. The drawback of the Y-K model and its reasons were also examined and investigated. In order to overcome this drawback, a new weighted

sum operator with an improved coefficient was provided to replace the Laplacian of u in the Y-K model. Numerical experiments showed that this improvement can not only keep the restored image clean but also reduces the generation of the speckles.

The fourth contribution of the study is to improve the Y-K model from the view of numerical implementation. Investigating the deeper reason of the speckles, an adaptive relaxation method was introduced into the numerical implementation. This can slow down the generation of speckles. Apart from this, a discontinuity detection and restoration around edges were also employed to improve the quality of the restored image. Results stated that these improvements can avoid the generation of speckles and enhance the quality of the restored image.

The fifth contribution is to develop a stopping criterion to help control the iterative procedure of image processing. Theoretical proof of the validity of the new criterion has not been investigated. However, by some numerical demonstrations, this criterion can be possible for controlling the iteration automatically.

The last contribution of the thesis is to investigate the variational methods used in image segmentation. The most important work was to explore and examine the active contour model. Behaviours of the active contour model as well as the choice of the coefficients in the internal energy term were addressed and explained. Based on such study, it was found that the existence of noise affects the segmentation result enormously. Out of this consideration and the advantage of the anisotropic diffusion model, an improved active contour model integrated with the anisotropic diffusion was proposed to achieve a better segmentation result even with the existence of noise.

8.2 Future work

1. The choice of the value of K will be further investigated aiming to propose an adaptive value of K for most of images.

2. The application of the new stopping criterion on other fourth order PDE methods will be made to check the validity of it.

3. A new vector flow is being investigated to explore the possibility to replace the external energy term in the active contour model. Level set methods will be introduced to solve the topology issue of the active contour model.

4. The issue of PSNR will be further examined to find out a new error measurement of the quality of an image.

References

[1] R. C .Gonzalez. Digital Image Processing (2nd Edition). Addison-Wesley Publishing Co. Inc. Boston, MA, USA, 1992.

[2] J. S. Lee. Digital Image Enhancement and Noise Filtering by Use of Local Statistics. *IEEE Transactions on Patter Analysis and Machine Intelligence*, **2**(2), 1980.

[3] S. G. Chang, B. Yu and M. Vetterli. Image Denoising via Lossy Compression and Wavelet Thresholding. *Proceedings of the IEEE International Conference Image Processing*, **1**, 1997.

[4] G. Gilboa, N. Sochen and Y. Y. Zeevi. Forward-and-Backward Diffusion Processes for Adaptive Image Enhancement and Denoising. *IEEE Transactions on Image Processing*, **11**(7), 2002.

[5] T. Kärkkäinen, K. Kunisch, K. Majava. Denoising of Smooth Images Using L1-Fitting. *Computing*, **74**(4), 2005.

[6] L. Shang and F. W. Cao. Adaptive Denoising Using a Modified Space Coding Shrinkage Method. *Neural Processing Letters*, **24**, 2006.

[7] D. Gabor. Information Theory in Electron Microscopy. Laboratory Investigation 14, 1965.

[8] A. K. Jain. Partial Differential Equations and Finite-difference Methods in Image Processing, Part 1: Image Representation. *Journal of Optimization Theory and Applications*, **23**, 1977.

[9] A. P. Witkin. Scale Space Filtering. *Int.Joint.Conf. Artificial Intelligence* **2**, 1983.

[10] J. J. Koenderink. The Structure of Images. Biological Cybernetics **50**, 1984.

[11] R. A. Hummel. Representations Based on Zero-crossing in Scale Space. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*,1986.

[12] P. Perona and J. Malik. Scale Space and Edge Detection Using Anisotropic Diffusion. *IEEE PAMI*, **12**, 1990.

[13] S. Kichenassamy. The Perona-Malik Paradox. *SIAM J. APPL. MATH.*, 57(5), 1997.

[14] M. J. Black, G. Sapiro, D. H. Marimont, et al. Robust Anisotropic Diffusion. *IEEE Transactions on Image Processing*, **7**(3), 1998.

[15] G. Gilboa, Y. Y. Zeevi, N. Sochen. Complex Diffusion Processes for Image Filtering. *Proceedings of the Third International Conference on Scale-Space and Morphology in Computer Vision*, 2001.

[16] J. Monteil and A. Beghdadi. A New Adaptive Nonlinear Anisotropic Diffusion for Noise Smoothing. *International Conference on Image Proceedings*, 1998.

[17] Y. Chen, B. C. Vemuri and L. Wang. Image Denoising and Segmentation via Nonlinear Diffusion. *Computers and Mathematics with Applications*, **39**, 2000.

[18] S. Esedoğul. An Analysis of the Perona-Malik Scheme. *Communications on Pure and Applied Mathematics*, **54**(12), 2001.

[19] L. I. Rudin, S.Osher and E. Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D*, **60**(1-4), 1992 .

[20] F. Catté, P. L. Lions and J. M. Morel, T. Coll. Image Selective Smoothing and Edge Detection by Nonlinear Diffusion I. *SIAM J. Numer. Anal.*, **29**, 1992.

[21] J. L. Morel and S. Solimini. Variational Methods in Image Segmentation. Birkhauser, Boston, 1995.

[22] P. Chen and Y. Wang. A New Fourth-order Equation Model for Image Inpainting. *6th International Conf. On Fuzzy Systems and Knowledge Discovery*, 2009.

[23] Y. L. You, M. Kaveh. Fourth-Order Partial Differential Equations for Noise Removal. *IEEE Transaction Image Processing*, **9**(10) , 2000.

[24] M. Lysaker, A. Lundervold and X. C. Tai. Noise Removal Using Fourth-order Partial Differential Equation with Applications to Medical Magnetic Resonance Images in Space and Time. *IEEE Transactions on Image Processing*, **12**(12), 2003.

[25] S. Kim and H. Lim. Fourth-order Partial Differential Equations for Effective Image Denoising. *Electronic Journal of Differential Equations*, **17**, 2009.

[26] M. R. Hajiaboli. An Anisotropic Fourth-Order Diffusion Filter for Image Noise Removal. *Int J Comput Vis*, 2010.

[27] F. Li, C. Shen, J. Fan and C. Shen. Image Restoration Combining a Total Variational Filter and a Fourth-order Filter. *J. Vis. Commun. Image R.*, **18**, 2007.

[28] R. Holm. Image Inpainting Using Nonlinear Partial Differential Equations. *Thesis in Applied Mathematics, University of Bergen*, 2005.

[29] D. Yi and S. Lee. Fourth-order Partial Differential Equations for Image Enhancement. *Applied Mathematics and Computation*, **175**, 2006.

[30] R. Haralick and R. Shapiro. Computer and Robot Vision, vol.1&2, Addison-Wesley Inc., Reading, Massachusetts, 1992.

[31] M. Kass, A. Witkin and D. Terzopoulos. Snakes: Active Contour Models. *International Conference on Computer Vision*, London, UK, 1987.

[32] Y. Fukada. Spatial Clustering Procedure for Region Analysis. *Patter Recognition*, **12**, 1980.

[33] T. Pavlidis and Y. Liow. Integrating Region Growing and Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(3), 1990.

[34] A. A. Amini, T. E. Weymouth, R. C. Jain. Using Dynamic Programming for Solving Variational Problem in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(9), 1990.

[35] Q. Faugora, R. Keriven. Variational Principles, Surface Evolution, PDE's Level Set Methods, and the Stereo Problem. *IEEE Transactions on Image Processing*, **7**(3), 1998.

[36] D. Cremers and C. Schonorr. Statistical Shape Knowledge in Variational Motion Segmentation. *Image and Vision Computing*, **21**(1), 2003.

[37] C. Xu and J. L. Prince. Snakes, Shapes, and Gradient Vector Flow. *IEEE Transactions on image Processing*, **7**(3), 1998.

[38] T. Chan and L. Vese. An Active Contour Model without Edges. *Lecture Notes in Computer vision*, **1682**, 1999.

[39] L. D. Cohen. On Active Contour Models and Balloons. *CVGIP: Image Understanding*, **53**(2), 1991.

[40] V. Caselles, R. Kimmel and G. Sapiro. Geodesic Active Contours. *5th International Conference on Computer Vision*, 1995.

[41] V. Caselles, F. Catté, T. Coll and F. Dibos. A Geometric Model for Active Contours in Image Processing. *Numer. Math.*, **66**(1), 1993.

[42] R. Malladi, J. A. Sethian and B. C. Vemuri. Shape Modelling with Front Propagation: a Level Set Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(2), 1995.

[43] B. B. Kimia, A. Tannenbaum and S. W. Zucker. Shapes, Shocks, and Deformations I: The Components of Two-Dimensional Shape and the Reaction-Diffusion Space. *International Journal of Computer Vision*, **15**(3), 1995.

[44] S. Osher and J. A. Sethian. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. Journal *of Computational Physics*, **79**, 1988.

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

[46] D. Peng, B. Merriman, S. Osher, H. Zhao and M. Kang.. A PDE-based Fast Local Level Set Method. *J. Comp. Phys.*, 155, 1999.

[47] C. Li, C. Xu, C. Gui and M. D. Fox. Level Set Evolution without Re-Initialization: A New Variational Formulation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[48] C. Li, C. Xu, C. Gui and M. D. Fox. Distance Regularized Level Set Evolution and Its Application to Image Segmentation. *IEEE Transactions on Image Processing*, **19**(12), 2010.

[49] T. Iijima. Basic Theory on Normalization of Pattern (in Case of a Typical One-Dimensional Pattern). *Bulletin of the Electrical Laboratory*, **26**, 1962.

[50] L. Alvarez, F. Guichard, P. L. Lions and J. M. Morel. Axioms and Fundamental Equations in Image Processing. *Arch. Rational Mech. Anal.*, **123**(3), 1993.

[51] T. Lindeberg. Scale-Space Theory: A Basic Tool for Analysing Structures at different Scales. *Journal of Applied Statistics*, **21**(2), 1994.

[52] T. Lindeberg. Scale-Space Theory in Computer Vision. *Kluwer Academic Publishers/Springer, Netherlands*, 1994.

[53] J. Babaud, A. P. Witkin, M. Baudin and R. O. Duda. Uniqueness of the Gaussian Kernel for Scale-Space Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 1986.

[54] P. T. Jackway and M. Deriche. Scale-Space Properties of the Multiscale Morphological Dilation-Erosion. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **18**, 1996.

[55] M. Nielsen, P. Johansen, O. F. Olsen and J. Weickert (eds). Scale-Space Theories in Computer Vision. *Lecture Notes in Computer Science, Springer*, 1999.

[56] A. Mackworth and F. Mokhtarian. Scale-Based Description and Regognition of Planar Curves and Two-Dimensional Shapes. *IEEE Transactions on Pattern Analysis and Machin Intelligence*, **8**, 1986.

[57] X. C. Tai, K. Morken, M. Lysaker and K. A. Lie. Scale Space and Variational Methods in Computer Vision. *Second International Conference, SSVM*, 2009.

[58] G. Aubert, P. Kornprobst. Mathematical Problems in Image Processing. *Springer*, New York, 2000.

[59] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier and F. Lenzen. Variational Methods in Imaging. *Springer*, New York, 2009.

[60] M. Marston. The calculus of Variations in Large. *American Mathematical Society*, 1934.

[61] R. Courant and D. Hilbert. Methods of Mathematical Physics (Vol. I). *Interscience Publishers*, 1953.

[62] N. I. Akhiezer. The Calculus of Variations. *Blaisdell Pub. Co., New York*, 1962.

[63] R. K. Nesbet. Variational Principles and Methods in Theoretical Physics and Chemistry. *Cambridge University Press*, UK, 2002.

[64] T. F. Chan, J. Shen and L. Vese. Variational PDE Models in Image Processing. *Notices of the AMS*, **50**(1), 2003.

[65] C. F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz and R. Kikinis. Processing and Visualization of Diffusion Tensor MRI. *Medical Image Analysis*, **6**, 2002.

[66] R. B. Buxton. Introduction to Functional Magnetic Resonance Imaging-Principles and Techniques. *Cambridge University Press*, Cambridge, UK, 2002.

[67] T. Acharya and A. K. Ray. Image Processing: Principles and Applications. *John Wiley and Sons, Inc.*, 2005.

[68] T. Chan and J. Shen. Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods. *SIAM*, 2005.

[69] T. Achyarya and A. K. Ray. Image Processing Principles and Applications. Published by John Wiley & Sons, Inc., New Jersey, 2005.

[70] P. P. Jonker. Morphological Image Processing: Architecture and VLSI Design. *Springer*, the Netherlands, 1992.

[71] P. Soille and M. Pesaresi. Advances in Mathematical Morphology Applied to Geoscience and Remote Sensing. *IEEE Trans. Geoscience and Remote Sensing*, **40**(9), 2002.

[72] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami and T. Shiose. A Novel Adaptive Morphological Approach for Degraded Character Image Segmentation. *Pattern Recognition*, **38**(11), 2005.

[73] C. Torrence and G. P. Compo. A Practical Guide to Wavelet Analysis. *Bulletin of the American Meteorological Society*, **79**(1), 1998.

[74] Y. Jin, E. D. Angelini and A. F. Laine. Wavelets in Medical Image Processing: Denoising, Segmentation, and Registration, Handbook of Medical Image Analysis: Advanced Segmentation and Registration Models, edited by J. Suri, D. L. Wilson and S. Laximinarayan. *Kluwer Academic Publishers*, New York, 2004.

[75] M. Schneider, P. Fiequth, W. C. Karl and A. S. Willsky. Multiscale Statistical Methods for the Segmentation of Signals and Images. *IEEE Trans. Image Processing*, **9**(3), 2000.

[76] S. W. Chee, R. M. Gray. Stochastic Image Processing. *Kluwer Academic/Plenum*, New York, 2004.

[77] R. M. Gray and L. D. Davisson. An Introduction to Statistical Signal Processing. *Cambridge University Press*, UK, 2004.

[78] Z. Lin, W. Zhang and X. Tang. Learning Partial Differential Equations for Computer Vision. *Microsoft Technical Report*, MSR-TR-2009-189, 2008.

[79] M. Bertalmio, A. L. Bertozzi and G. Sapiro. Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting. *Proceeding of CVPR'01*, 2001.

[80] M. Bertalmio. Processing of Flat and Non-Flat Image Information on Arbitrary Manifolds Using Partial Differential Equations. *PhD Thesis*, *University of Minnesota*, 2001.

[81] F. Voci, s. Eiho, N. Sugimoto and H. Sekiguchi. Estimating the gradient in the Perona-Malik equation. *IEEE Signal Processing Magazine*, 23(3), 2004.

[82] S. Roman. Advanced Linear Algebra (third edition). *Springer*, New York, 2000.

[83] D. Marr and E. Hildreth. Theory of Edge Detection. *Proc. R. Soc. Lond. B, Biological Science*, **207**, 1989.

[84] B. M. ter Haar Romeny, L. Florack, J. J. Koenderink and M. A. Viergever. Scale Space: Its Natural Operators and Differential Invariants. *Proceedings of the 12th International Conference on Information Processing in Medical Imaging*, 1991.

[85] H. Ma îre. Modern Digital Image Processing. *Publishing House of Electronic Industry*, Beijing, 2006.

[86] Y. L. You, M .Kaveh, W. Y. Xu and A. Tannenbaum. Analysis and Design of Anisotropic Diffusion for Image Processing. *ICIP, IEEE international Conference*, 1994.

[87] X. Li, T. Chen. Nonlinear Diffusion with Multiple Edginess Thresholds. *Pattern Analysis and Machine Intelligence*, **7**(3), 1994.

[88] Y. -L. You, W. Xu, A. Tannenbaum, M.Kaveh. Behavioral Analysis of Anisotropic Diffusion in Image Processing. *IEEE Transactions on Image Processing*, **5**(11), 1996.

[89] S. Didas, J. Weickert and B. Burgeth. Stability and Local Feature Enhancement of Higher Order Nonlinear Diffusion Filtering. Berlin: *Pattern Recognition, Lecture Notes in Computer Science, Springer*, **3663**, 2005.

[90] J. Weickert. Anisotropic Diffusion in Image Processing. *B.G.Teubner Stuttgart*, 1998.

[91] M. J. Black, G. Sapiro, D. H. Marimont, D. Hegger. Robust Anisotropic Diffusion. *IEEE Transactions on Image Processing*, **7**(3), 1998.

[92] G. Gilboa, Y.Y. Zeevi. N. Sochen. Anisotropic Selective Inverse Diffusion for Signal Enhancement in the Presence of Noise. *IEEE International Conference*, 2000.

[93] D. Zwillinger. Handbook of Differential Equations (3rd Edition). *Academic Press, Florida*, 1997.

[94] P. Charbonnier, M. Barlaud, G. Aubert. Deterministic Edge-Preserving Regularization in Computed Imaging. *IEEE international Conference on Image Processing*, **2**(6), 1997.

[95] R. Glowinski. Numerical Methods for Nonlinear Variational Problems. *Springer*, 1984.

[96] R. C. Gonzelaz and R. E. Woods. Digital Image Processing (2nd Edition). *Addison-Wesley Longman Publishing co., Inc., Boston, Ma,* 1992.

[97] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. Numerical Recipes 3rd Edition: The Art of Scientific Computing. *Cambridge University Press*, 2007.

[98] Z. Lin and Q. Shi. An Anisotropic Diffusion PDE for Noise Reduction and Thin Edge Preservation. *Proceedings of the 10th International Conference on Image Analysis and Processing*, 1999.

[99] S. K. Weeratunga and C. Kamath. An Investigation of Implicit Active Contours for Scientific Image Segmentation. *Visual Communications and Image Processing Conference*, 2003.

[100] Y. Yagi, M. Brady, Y. Kawasaki and M. Yachida. Active Contour Road Model for Road Tracking and 3D Road shape Reconstruction. *Electronics and Communications in Japan*, **88**(9), 2005.

[101] S. Huq, B. Abidi, A. Goshtasby and M. Abidi. Stereo Matching with Energy Minimizing Snake Grid for 3D Face Modeling. *Biometric Technology for Human Identification*, **5404**, 2004.

[102] Y. Yagi, M. Brady, Y. Kawasaki and M. Yachida. Active Contour Road Model for Road Tracking and 3D Road Shape Reconstruction. *Electronics and Communication in Japan*, **88**(9), 2005.

[103] K. Washizu. Variational Methods in Elasticity and Plasticity. Pergamon Press, London, 1982.

[104] L. D. Cohen and I. Cohen. Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(11), 1993.

[105] D. Geiger, L. A. Costa and J. Vlontzos. Dynamic Programming for Detecting, Tracking and Matching Deformable Contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(3), 1995.

[106] L. Ji and H. Yan. Attractable Snakes Based on the Greedy Algorithm for Contour Extraction. *Pattern Recognition*, **35**(4), 2002.

Appendix

The codes using in this thesis were mostly programmed by the author except the TV model which was provided by the website http://www.math.ucla.edu/~lvese/285j.1.05s/TV_L2.m. The code of ACM model was referred to the website ftp://ftp.gentoo.diku.dk/diku/image/publications/nikolas.070901.pdf and then revised by the author. The different PDE-based denoising codes are based on a similar scheme, therefore, the code for the P-M model is provided here for reference, which includes image operation two files related to the operation on an image and main functions of the code.

BmpOperator.h

```
#ifndef MY_BMPOPERATOR
#define MY_MBPOPERATOR
#include <fstream>
#include <cstdlib>
#include <afx.h>
using namespace std;
int readWidth(CString filename);//read the width of an image
int readHeight(CString filename);// read the height of an image
bool bmpRead(unsigned char **image, CString filename);//read an image
bool bmpWrite(unsigned char **image, CString filename);//store an image
#endif
```

BmpOperator.cpp

```
#include "bmpOperator.h"
#include "StdAfx.h"
BITMAPFILEHEADER bmf;
BITMAPINFOHEADER bmi;
RGBQUAD *rgbQuad = NULL;
```

```
int offbits;
int nWidth;
int nHeight;
int paletteNum;
int nWidth_4;
int readWidth(CString filename)
{
	CFile bmpFile;
	try
	{
		bmpFile.Open(filename,CFile::modeRead);
		bmpFile.Read(&bmf,sizeof(BITMAPFILEHEADER));
		bmpFile.Read(&bmi, sizeof(BITMAPINFOHEADER));
		return (bmi.biWidth+3)/4*4;
		bmpFile.Close();
	}
	catch(CFileException *e)
	{
		return e->m_cause;
	}
}
int readHeight(CString filename)
{
	CFile bmpFile;
	try
	{
		bmpFile.Open(filename,CFile::modeRead);
		bmpFile.Read(&bmf,sizeof(BITMAPFILEHEADER));
		bmpFile.Read(&bmi, sizeof(BITMAPINFOHEADER));
		return bmi.biHeight;
		bmpFile.Close();
	}
	catch(CFileException *e)
	{
		return e->m_cause;
	}
}
bool bmpRead(unsigned char **image, CString filename)
{
	CFile bmpFile;
```

```
	int i = 0;
	try
	{
		bmpFile.Open(filename,CFile::modeRead);
		bmpFile.Read(&bmf, sizeof(BITMAPFILEHEADER));
		bmpFile.Read(&bmi, sizeof(BITMAPINFOHEADER));
		if(bmi.biClrUsed == 0)
			paletteNum = 1 << bmi.biBitCount;
		else
			paletteNum = bmi.biClrUsed;
		rgbQuad = new RGBQUAD[paletteNum * sizeof(RGBQUAD)];
		bmpFile.Read(rgbQuad,sizeof(RGBQUAD) * paletteNum);
		offbits = bmf.bfOffBits;
		nWidth = bmi.biWidth;
		nHeight = bmi.biHeight;
		nWidth_4 = (nWidth+3)/4*4;
		bmpFile.Seek(offbits, CFile::begin);
		for( i = 0; i < nHeight; i++)
		{
			bmpFile.Read(image[nHeight - 1 - i], sizeof(unsigned char)*nWidth_4);
		}
		bmpFile.Close();
	}
	catch(CFileException *e)
	{
		CString str;
		str.Format("the reason for failed reading:%d", e->m_cause);
	}
	return TRUE;
}

bool bmpWrite(unsigned char **image, CString filename)
{
	CFile bmpFile;
	int i = 0;
	try
	{
		bmpFile.Open(filename, CFile::modeWrite|CFile::modeCreate);
		bmpFile.SeekToBegin();
		bmpFile.Write(&bmf, sizeof(BITMAPFILEHEADER));
```

```
                bmpFile.Flush();
                bmpFile.SeekToEnd();
                bmpFile.Write(&bmi, sizeof(BITMAPINFOHEADER));
                bmpFile.Flush();
                bmpFile.SeekToEnd();
                bmpFile.Write(rgbQuad, sizeof(RGBQUAD)*paletteNum);
                bmpFile.Flush();
                bmpFile.Seek(offbits,CFile::begin);
                for(i = 0; i < nHeight; i++)
                {
                        bmpFile.Write(image[nHeight -1 -i],sizeof(unsigned char)*nWidth_4);
                }
                bmpFile.Close();
        }
        catch(CFileException *e)
        {
                CString str;
                str.Format("the reason for failed write:%d", e->m_cause);
        }
        return TRUE;
}
/*
```

Function: OnProcess()

Argument: none

Return: none

Description: the main iterative process of the P-M model

```
*/

void CProcessDlg::OnProcess()
{
        int  i,j;  //indices point to the position of an image
        int count=0,t;//iteration number
        CString msg;
        double k = 10; //Value of K in the coefficient
        double l=0.25;//time step
        UpdateData(true);
```

```
if (m_done)
{
	CDialog::OnOK();
	return;
}
//allocate memory for the noisy image and denoised image
m_width = readWidth(m_noise_file);
m_height = readHeight(m_noise_file);
m_noise_image = new unsigned char *[m_height];
for(i = 0; i < m_height; i++)
{
	m_noise_image[i] = new unsigned char[m_width];
}
result_image = new unsigned char *[m_height];
for(i = 0; i < m_height; i++)
{
	result_image[i] = new unsigned char[m_width];
}
if(!bmpRead(m_noise_image, m_noise_file))
	return;
//allocate  memory for the smooth image
m_smooth_image = new double *[m_height];
for(i = 0; i < m_height; i++)
{
	m_smooth_image[i] = new double[m_width];
}
//allocate memory for the temp image
m_temp_image = new double *[m_height + 2];
for(i = 0; i < m_height + 2; i++)
{
	m_temp_image[i] = new double[m_width + 2];
}
//allocate memory for the gradient
u_gradient_image = new double *[m_height];
for(i = 0; i < m_height; i++)
{
	u_gradient_image[i] = new double[m_width];
}
 //allocate memory for the coefficient function
m_g = new double *[m_height];
```

```
for(i = 0; i < m_height; i++)
{
        m_g[i] = new double[m_width];
}
 // initilisation
for(i=0;i<m_height+2;++i)
        for(j=0;j<m_width+2;j++)
                m_temp_image[i][j] = 0;
for( i=0; i<m_height; ++i)
        for( j=0; j<m_width; ++j)
        {
                m_smooth_image[i][j] = (double)m_noise_image[i][j];
                m_temp_image[i+1][j+1] = m_smooth_image[i][j];
        }
//the iterative process
for(t=0;t<m_iterations;t++)
{
        ++count;
        for(j=1;j<m_width+1;j++)
        {
                m_temp_image[m_height+1][j] = (double) m_temp_image[m_height][j];
                m_temp_image[0][j] = (double) m_temp_image[1][j];
        }
        for(i=1;i<m_height+1;i++)
        {
                m_temp_image[i][m_width+1] = (double) m_temp_image[i][m_width];
                m_temp_image[i][0] = (double) m_temp_image[i][1];
        }
        for(i=0;i<m_height;i++)
                for(j=0;j<m_width;j++)
                         m_smooth_image[i][j] = m_temp_image[i+1][j+1]
                        +l*(g(m_temp_image[i+2][j+1]
                        -m_temp_image[i+1][j+1],k)*(m_temp_image[i+2][j+1]
                        -m_temp_image[i+1][j+1])
                         +g(m_temp_image[i][j+1]
                        -m_temp_image[i+1][j+1],k)*(m_temp_image[i][j+1]
                        -m_temp_image[i+1][j+1])
                        +g(m_temp_image[i+1][j+2]
                        -m_temp_image[i+1][j+1],k)*(m_temp_image[i+1][j+2]
```

```
                                    -m_temp_image[i+1][j+1])
                                   +g(m_temp_image[i+1][j]
                                  -m_temp_image[i+1][j+1],k)*(m_temp_image[i+1][j]
                                  -m_temp_image[i+1][j+1]) );
                for(i=1;i<m_height+1;i++)
                        for(j=1;j<m_width+1;j++)
                        m_temp_image[i][j] = m_smooth_image[i-1][j-1];
                msg.Format("interations %d.", count);
                 Message(msg);
          }
          msg.Format("interations %d.", count);
          Message(msg);
          // store the denoised image
         for(i = 0; i < m_height; i++)
         {
                for (j = 0; j < m_width; j++)
                {
                        result_image[i][j] = (unsigned char)m_smooth_image[i][j];
                }
         }
         if(!bmpWrite(result_image, m_smooth_file))
                return;
         m_ProcessButton.SetWindowText("Close");
        m_done=true;
}
/*
```

Function: g(double intensity, int k)

Arguments: intensity – the intensity of position (i, j)

k – the value of K in the coefficient

Return: the value of coefficient in position (i, j)

Description: the coefficient in the P-M model

```
*/

double CProcessDlg::g(double a,int k)
{
        double result;
```

```
            result = 1/(1+pow(a/k,2));//p-M
//          result = 2/sqrt(1+pow(a,2));//minimal surfaces
//          result = exp(-pow(a,2)/pow(k,2));//the exponential coefficient by P-M model
//          result = 1;//isotropic diffusion
            return result;
}
/*
```

Function: u_gradient(int i, int j)

Arguments: (i, j) – the indices to the position

Return: the gradient at position (i, j)

Description: calculate the gradient of the image

```
*/

double CProcessDlg::u_gradient(int a, int b)
{
            double result,tempx,tempy;
            int i = a + 1, j = b + 1;
            tempx = m_temp_image[i+1][j] - m_temp_image[i][j];
            tempy = m_temp_image[i][j+1] - m_temp_image[i][j];
            result = sqrt(pow(tempx,2)+pow(tempy,2));
            return result;
}
```