Variational and PDE-based methods for image processing

Jinming Duan

School of Computer Science
University of Nottingham

This dissertation is submitted for the degree of
Doctor of Philosophy

July 2018

Acknowledgements

Firstly, I am very grateful to my supervisor Dr Li Bai, for her constant encouragement and support over the past three years. She is always full of kindness, very happy to share her academic ideas and life experiences, and super fast to read and polish my manuscripts. She has given me not only the cutting-edge research topics, but also the freedom to do what I would like to do.

I would like to thank the University of Nottingham for funding my PhD research. Many thanks also go to all of my lab mates in the Computational Imaging and Graphics team. These are Ben Haines, Wil OC Ward, Yuchun Ding, Luke Sibbett and Faiza Bukenya, who have given me a great amount of support both in academia and in everyday life.

Finally, I am extremely grateful to my grandfather Chuanwen Duan, my mother Yaping Zhang, my father Jiaqi Duan, my beloved wife Wenqi Lu, and a lot of other lovely members in my family, for their endless love and support over all these years!

Abstract

In this thesis, we study modern variational and partial differential equation (PDE)-based methods for three image analysis applications, namely, image denoising, image segmentation, and surface reconstruction from point clouds. A common feature these applications have is the use of novel variational formulations.

For image denoising, we focus on higher order variational functionals in which the regulariser incorporates second order derivatives or is a sophisticated combination of first and second order derivatives. We study seven representative first and/or second order functionals, implement them using the efficient split Bregman algorithm, and compare their performances. With the knowledge of the main properties of each of the denoising approaches, we can then select and adapt them for image segmentation.

For image segmentation, we are in particular interested in images of three types: red blood cell (RBC) images, histology images of the microglial cells, and optical coherence tomography (OCT) images of the retina. For RBC images we develop an automated and accurate image analysis framework for an image-based cytometer that uses variational total generalised variation, adaptive thresholding and support vector machine. The framework can 1) detect and numerically count malaria parasite infected RBCs acquired from Giemsa-stained smears; 2) classify all parasitic subpopulations by quantifying the area occupied by the parasites within the infected cells; 3) predict if the RBC image has been infected by malaria parasites. We show the effectiveness of the framework by quantifying and classifying both RBC and infected RBC images.

For histology images of the microglial cells, we introduce an automated image segmentation method that is capable of efficiently extracting microglial cells from the images. The method uses variational Mumford-Shah total variation and split Bregman for image denoising and segmentation and is fast, accurate and robust against noise and inhomogeneity in the image. We evaluate the method on the image data from wild type mice and transgenic mouse models of Alzheimer's disease. The method is scalable to large datasets, allowing microglia analysis in regions of interest and across the whole brain.

For OCT images of the retina, we propose a novel and accurate geodesic distance method to segment healthy and pathological OCT images, in both two and three

dimensions. The method uses a weighted geodesic distance by an exponential function, taking into account horizontal and vertical intensity variations. The fast sweeping method is used to derive the geodesic distance from an Eikonal equation, a special case of Hamilton-Jacobi equations that belongs to the family of nonlinear PDEs. Segmentation is then achieved by solving an ordinary differential equation using the resulting geodesic distance. The proposed method is also extensively compared with the parametric active contour model and graph theoretic methods.

Finally, we study surface reconstruction from point clouds. We treat this reconstruction problem as an image segmentation problem and hence develop a novel variational level set method. Th method is capable of reconstructing implicit surfaces from unorganised point clouds while preserving fine details of the surfaces. A distance function, derived from the point cloud using the fast sweeping algorithm, is used as an edge indicator function and to find an initial image enclosed by the point cloud. A novel variational segmentation functional is then proposed that effectively integrates the initial image and edge indicator. Gradient descent optimisation finally minimises the functional and ensures an accurate and smooth reconstruction.

Table of contents

Related publications

Journal papers

1. W Lu, **J Duan**, Z Qiu, Z Pan, R Liu and L Bai. Implementation of high order variational models made easy for image processing. Mathematical Methods in the Applied Sciences, 39 (14), 4208-4233. 2016

2. D Yang, G Subramanian, **J Duan**, S Gao, L Bai, R Chandramohanadas and Ye Ai. A portable image-based cytometer for rapid Malaria detection and quantification. PLoS ONE 12.6 (2017): e0179161.

3. Y Ding, M Pardon, A Agostini, H Faas, **J Duan**, W Ward, F Easton, D Auer and L Bai. Novel methods for microglia segmentation, feature extraction and classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 14 (6), 1366-1377, 2017.

4. **J Duan**, C Tench, I Gottlob, F Proudlock and L Bai. New variational image decomposition model for simultaneously denoising and segmenting optical coherence tomography images. Physics in Medicine and Biology, 60 (22): 8901-8922, 2015.

5. **J Duan**, C Tench, I Gottlob, F Proudlock and L Bai. Automated segmentation of retinal layers from optical coherence tomography images using geodesic distance. Pattern Recognition, 72 (2017): 158–175.

Conference papers

1. **J Duan**, B Haines, W Ward and L Bai. Surface reconstruction from point clouds using a novel variational model. Research and Development in Intelligent Systems XXXII 135-146, 2015.

2. **J Duan**, C Tench, I Gottlob, F Proudlock and L Bai. Optical coherence tomography image segmentation. IEEE International Conference on Image Processing (ICIP), 2015: 4278-4282.

3. **J Duan**, Y Ding, Z Pan, J Yang and L Bai. Second order Mumford-Shah model for image denoising. IEEE International Conference on Image Processing (ICIP), 2015: 547-551.

List of figures

List of tables

List of tables

Chapter 1

Introduction

1.1 Digital images and their mathematical representation

In today's society, we almost encounter digital images everyday, from photography in our daily life to medical imaging, remote sensing, astronomical imaging, etc. For example, microscopy images are used in visualising biological processes, observing cell structure, studying cancers at molecular level, and detecting the smallest items of evidence for forensic sciences; medical images are used in revealing internal structures hidden by the skin and bones, diagnosing and treating diseases, identifying abnormalities, and testing the effectiveness of medical treatment and therapy; satellites images are used in tracking earth resources and geographical mapping, studying agricultural crops and urban population, forecasting weather, and controlling flood and fire; astronomical images are used in studying stars, planets and space.

However, the images produced in these scenarios usually have to be processed so as to gain more meaningful information. The pre-processing of digital images is a common task in image processing or computer vision. Since images in reality are often suffering from noise, blur, or incompleteness, it is imperative to develop image processing methods that are capable of improving the quality of these images. Examples of incomplete images can come from medical imaging such as magnetic resonance imaging (MRI) of the brain, positron emission tomography (PET) imaging of the heart, and X-ray imaging of human bones. These imaging devices usually introduce noise to the image. Other examples include satellite images of our earth and telescope images of the universe, which are often blurred. In summary, effective digital image restoration methods are needed to remove noise and/or blur from the image or to recover lost information.

Image segmentation simplifies and/or changes the representation of an image into something that is more meaningful and easier to analyse. For example, if we are interested in certain objects in an image and then employ a segmentation algorithm to exact them out of it, the rest of content in that image can be completely ignored. The objects extracted are normally used to derive certain statistics for high-level image analysis and understanding. Finally, there is also an important issue of organising image data in a meaningful way. To do so, we have to think about image classification. Classification is to assign new date samples to target categories or classes. The ultimate goal of image classification is to accurately predict the target class for each sample in a new dataset. It is a very powerful tool for digital image analysis. There are also many other image processing applications we will not discuss here. We however refer to [3] for a more comprehensive introduction to digital image processing.

Although there exist a great amount of image processing applications, there are still problems in this area that have not yet been fully and satisfactorily resolved. It is not surprising, therefore, that this is still a very active research field. From mathematicians, statistics, engineers to computer scientists, a large group of people have been and are still working in this area.

This thesis is concerned with modern techniques that use variational methods and partial differential equations (PDEs) for different image analysis problems. Before we go deeper into the methodologies, we need to first understand what an image, or more specifically a digital image, really is. Generally speaking, a digital image, acquired from some imaging device (i.e. microscope, digital camera, medical scanner, satellite, telescope, etc.), normally represents a certain scene in the continuous universe. By sampling the image acquisition device first establishes a discrete regular grid on the scene and then assigns a value (e.g. the brightness) to each grid element (quantisation). These grid elements are often referred to as pixels in two dimension (2D) or voxels in three dimension (3D). Depending on the image acquisition device used, the image content and its value range for each pixel/voxel are different. For example, the image content can be brightness (grayscale) values that are scalar values ranging from $[0, 255]$ in an 8-bit imaging device or $[0, 2^{32} - 1]$ in a 32-bit imaging device.

Let Ω be an open bounded subset of $\mathbb{R}^n$ and $x \in \Omega$, the mathematical representation of a digital image f is

$$f(x) : (\Omega \subset \mathbb{R}^n) \to \mathbb{R}^d, \tag{1.1}$$

where $n = 2$ and $n = 3$ respectively correspond to the 2D and 3D images. $d = 1$ denotes the real valued grayscale images, while $d > 1$ represents the multispectral (hyperspectral) images. For example, an RGB colour image can be represented in the case when $d = 3$.

(1.1) is the continuous representation of a digital image. The discrete representation of a typical image is the form of

$$f : \{1, ..., N\} \times \{1, ..., M\} \to \{0, ..., 255\}^3. \tag{1.2}$$

Such a discrete representation in (1.2) indicates the image is an RGB colour image defined on a 2D rectangular grid with resolution $N \times M$. Analogically, a discrete grayscale image defined on a 3D box-shaped grid with resolution $N \times M \times H$ is given by

$$f : \{1, ..., N\} \times \{1, ..., M\} \times \{1, ..., H\} \to \{0, ..., 255\}. \tag{1.3}$$

For the sake of clarity, in Figure 1.1 we visualise the connection between a discrete digital image and its image function f for the case when $n = 2$ and $d = 1$ in (1.1). We note that in order to apply the mathematical theory of variational and PDE-based methods, the derivation and analysis is first done in the continuous setting and then returning to the discrete setting for numerical computation.

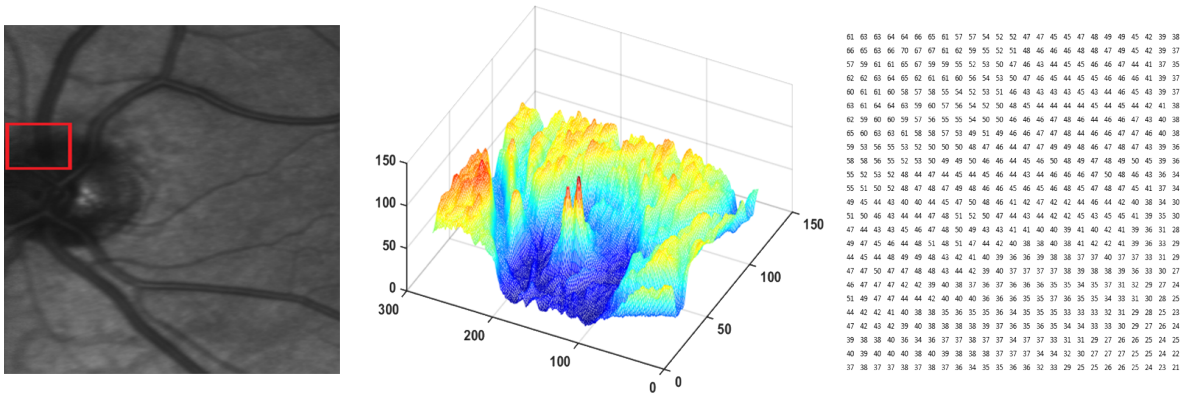


Figure 1.1: Digital image and image function. Left: an en-face retinal image; Middle: the function of the image where the grayvalue f is plotted against the x, y coordinates; Right: matrix from the grayvalues highlighted in the red rectangle in the left image.

Since we can represent a digital image with a mathematical function, it is then straightforward for us to apply mathematical operations to manipulate it. There exist a large number of well-established mathematical theories that can be used to process an image. Popular techniques range from statistical methods, wavelet methods, morphological operators, combinatorial optimisation, variational and PDE-based methods. In this thesis, we are particularly interested in the variational and PDE-based methods. We will develop and implement these methods for different imaging applications such as image denoising, image segmentation and surface reconstruction. In the following sections, we

shall first introduce the variational and PDE-based methods, which are central to the image analysis applications in the later chapters.

1.2 Variational methods

Mathematicians use the term 'energy functional' or 'functional' to describe an object that takes a function as an input and gives another function as an output. In such a case, 'variables' within the object are functions rather than the conventional variables represented by generic symbols such as x and y. The main idea of variational methods is to find the solution of a problem through minimising a certain energy functional based on certain assumptions on the problem solution. Let us consider the following general functional with derivatives up to the second order

$$E(u) = \int_{\Omega} \mathcal{L}\left(x, u(x), \nabla u(x), \nabla^2 u(x)\right) dx. \tag{1.4}$$

Here $u(x)$ is the function to be minimised over. $\mathcal{L}$ is the Lagrangian which represents the assumptions and is associated with input image function f as defined in (1.1). ∇u and $\nabla^2 u$ represent the gradient vector and the Hessian operator which are respectively defined in (1.5) and (1.6)

$$\nabla u(x) = \left(\frac{\partial u}{\partial x_1}, ..., \frac{\partial u}{\partial x_n}\right) \in \mathbb{R}^n, \tag{1.5}$$

$$\nabla^2 u(x) = \begin{pmatrix} \frac{\partial^2 u}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 u}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 u}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 u}{\partial x_n \partial x_n} \end{pmatrix} \in \mathbb{R}^{n \times n}. \tag{1.6}$$

These differential operators are expressed in a general form so they can represent arbitrary dimension (depending on n). For compact notations, we often abbreviate $\frac{\partial u}{\partial x_i}$ as u_{x_i} or $\partial_{x_i} u$ and $\frac{\partial^2 u}{\partial x_i \partial x_j}$ as $u_{x_i x_j}$ or $\partial_{x_i} \partial_{x_j} u$. For a 2D image case, $x_1 = x$ and $x_2 = y$. The order of derivatives in $\mathcal{L}$ may vary depending on the requirements of the problem. The energy functional E in (1.4) is practically meaningful once its minimiser with respect to u is found. This minimisation process inevitably entails the calculus of variations, which we discuss in the next section.

1.2.1 Calculus of variation

Calculus of variations is a mathematical tool that deals with the problem of optimising functionals. Such a tool offers a necessary condition called Euler-Lagrange equations which are the resulting formulations by minimising a given functional associated with multiple function variables. The role of this criterion is similar to that of the first order necessary condition for optimality in standard differential calculus, that is, the first order derivatives with respect to variables of a given function must vanish at extrema. Analogously to the classical differential calculus, if the given energy functional is strictly convex, a unique minimiser may be able to find. In contrast, if the functional is non-convex, it may have multiple local extrema. Sometimes, the functional may not has a solution or even if it has a solution it may be hard to identify if the solution is global.

Let us now present the Euler-Lagrange equation of the general form (1.4), which can be derived by integration by parts and the Gauss's theorem along with the fundamental lemma in calculus of variation. Thus, for an equivalent energy functional of (1.4) in 2D cases

$$E(u) = \int_{\Omega} \mathcal{L}(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yx}, u_{yy})\, dxdy, \tag{1.7}$$

we have its Euler-Lagrange equation that is the form of

$$\mathcal{L}_u - \frac{\partial}{\partial x}\mathcal{L}_{u_x} - \frac{\partial}{\partial y}\mathcal{L}_{u_y} + \frac{\partial^2}{\partial x^2}\mathcal{L}_{u_{xx}} + \frac{\partial^2}{\partial x \partial y}\mathcal{L}_{u_{xy}} + \frac{\partial^2}{\partial y \partial x}\mathcal{L}_{u_{yx}} + \frac{\partial^2}{\partial y^2}\mathcal{L}_{u_{yy}} = 0,$$

with the natural boundary conditions imposed on the boundary $\partial\Omega$

$$\boldsymbol{n}^{\mathrm{T}} \begin{bmatrix} \mathcal{L}_{u_x} - \frac{\partial}{\partial x}\mathcal{L}_{u_{xx}} - \frac{\partial}{\partial y}\mathcal{L}_{u_{xy}} \\ \mathcal{L}_{u_y} - \frac{\partial}{\partial x}\mathcal{L}_{u_{yx}} - \frac{\partial}{\partial y}\mathcal{L}_{u_{yy}} \end{bmatrix} = 0,$$

where $\boldsymbol{n}$ is an outer normal vector $(n_1, n_2)^{\mathrm{T}}$. Because (1.7) contains the second order differential operator, the following conditions must be further imposed

$$\boldsymbol{n}^{\mathrm{T}} \begin{bmatrix} \mathcal{L}_{u_{xx}} \\ \mathcal{L}_{u_{xy}} \end{bmatrix} = 0 \quad \text{and} \quad \boldsymbol{n}^{\mathrm{T}} \begin{bmatrix} \mathcal{L}_{u_{yx}} \\ \mathcal{L}_{u_{yy}} \end{bmatrix} = 0.$$

1.2.2 Regularisation

So far, we have shown how to compute the Euler-Lagrange equation of a functional using calculus of variations. We now consider to integrate regularisation and energy functional for general image processing applications. The variational framework we have introduced above is a natural way of performing this integration.

The nature of most image processing problems is often ill-posed, meaning that there may not exist a solution for these problems. In order to alleviate the ill-posedness, the regularisation process must be adopted. The underlying theory behind regularisation is that the ill-posed problem may be reformulated in a way such that an appropriate solution to the reformulated problem can be found. Such solution is usually sufficient to approximate the original problem in an admissible level. Furthermore, the reformulation process can be improved by imposing some constraints such as smoothness (regularity) on the given problem. In other words, the conditioning of the given problem can be improved by imposing smoothness. Afterwards, a desirable numerical solution can be obtained by minimising the regularised energy functional using the calculus of variation introduced in the last section.

In the context of image analysis, a functional normally has a data fidelity term and a regularisation/smoothness term. Therefore, one may represent general image processing problems with the following energy functional

$$E(u) = \int_{\Omega} \mathcal{D}_{f(x)}(u(x))dx + \alpha \int_{\Omega} \mathcal{R}(u(x))dx, \tag{1.8}$$

where $f(x) : (\Omega \subset \mathbb{R}^n) \to \mathbb{R}^d$ ($n = 2$ and $d = 1$ in this thesis) is the input image function. $\int_{\Omega} \mathcal{D}_{f(x)}(u(x))dx$ denotes the data fidelity term associated with f. It has different forms depending on applications. For example, in image denoising it is formulated according to the noise type. $\mathcal{D}_{f(x)}(u(x)) = (u(x) - f(x))^2$ if $f(x)$ is corrupted by Gaussian white noise, or $\mathcal{D}_{f(x)}(u(x)) = |u(x) - f(x)|$ if $f(x)$ is corrupted by impulsive noise. $\int_{\Omega} \mathcal{R}(u(x))dx$ is the regularisation term, and α is a positive regularisation parameter. Technically, the robust behaviour of variational methods is in fact attributed to the smoothness term $\int_{\Omega} \mathcal{R}(u(x))dx$ that is capable of taking into account information of neighbouring data points. It is worth mentioning that this feature is particularly useful especially when the image contains missing information and noise. In this context, the regularisation parameter plays a role in controlling the level of smoothness of the solution. We will show different definitions in detail for $\mathcal{R}$ in Chapter 2.

1.3 PDE-based methods

A PDE is an equation that involves functions as variables and their partial derivatives. PDEs arise in many areas of physical science, since they are very effective in modelling a wide variety of phenomena such as sound, heat, electrostatics, electrodynamics, fluid dynamics, elasticity, or quantum mechanics. Throughout the thesis, we will experience

two types of PDEs for image processing and analysis, including linear and nonlinear cases.

1.3.1 Second and fourth order linear PDEs

For the first type of PDEs, we are interested in the following second order linear equation

$$u(x) - \theta \Delta u(x) = g(x). \tag{1.9}$$

Here, $g(x) : (\Omega \subset \mathbb{R}^n) \to \mathbb{R}^d$, $x \in \Omega$ ($n = 2$ and $d = 1$ in this thesis), which is associated with the input image $f(x)$ as defined in (1.1). u is the function to be calculated. Δ is the Laplace operator or Laplacian which is also denoted as $\Delta = \nabla \cdot (\nabla u) = div(\nabla u)$, where ∇ is the gradient operator in (1.5) mapping scalar functions to vector functions and $\nabla\cdot$ is the divergence operator (also symbolised "div") mapping vector functions to scalar functions. The divergence of a vector function, i.e.,

$$\boldsymbol{p} = (p_1, ..., p_n) \in \mathbb{R}^n,$$

is given by

$$div\left(\boldsymbol{p}\right) = \partial_{x_1} p_1 + \cdots + \partial_{x_n} p_n. \tag{1.10}$$

We note that the divergence notations $\nabla\cdot$ and div are used interchangeably in the thesis. With the divergence and gradient at hand, it is easy to check that

$$\Delta u = \partial_{x_1}\partial_{x_1} u + \cdots + \partial_{x_n}\partial_{x_n} u. \tag{1.11}$$

In (1.9), u and g are real valued functions on an image grid, and g is given and u is sought. θ is a positive coefficient resulted from applying a variable splitting algorithm for an image processing problem. We shall see how this parameter arise in Chapter 2. (1.9) is said to be second order linear because the Laplacian in (1.11) is a second order linear differential operator and after (1.9) is fully discretised on an image grid the resulting equations are a linear system. Boundary condition also needs to be considered for (1.9). To benefit from the discrete fast Fourier transform (FFT) solver, which leads to a closed-form solution of u, we apply the periodical boundary condition for (1.9) in the thesis. See Appendix A for the detailed implementation of (1.9) with the FFT.

A fourth order linear PDE is also studied, which has the form of

$$u\left(x\right) + \theta div^2\left(\nabla^2 u\left(x\right)\right) = g\left(x\right), \tag{1.12}$$

where θ and g have the same meaning as those in (1.9). ∇^2 is the Hessian operator defined in (1.6), and div^2 represents the second order divergence operator. If it is applied to a matrix valued function

$$\boldsymbol{q} = \begin{pmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{pmatrix} \in \mathbb{R}^{n \times n},$$

we have

$$div^2(\boldsymbol{q}) = \partial_{x_1}\partial_{x_1}q_{11} + \cdots + \partial_{x_1}\partial_{x_n}q_{1n} + \cdots + \partial_{x_n}\partial_{x_1}q_{n1} + \cdots + \partial_{x_n}\partial_{x_n}q_{nn}. \tag{1.13}$$

Hence, one can easily check that

$$\begin{aligned} div^2\left(\nabla^2 u\right) = {} & \partial_{x_1}\partial_{x_1}\partial_{x_1}\partial_{x_1}u + \cdots + \partial_{x_1}\partial_{x_n}\partial_{x_1}\partial_{x_n}u + \cdots \\ & + \partial_{x_n}\partial_{x_1}\partial_{x_n}\partial_{x_1}u + \cdots + \partial_{x_n}\partial_{x_n}\partial_{x_n}\partial_{x_n}u. \end{aligned} \tag{1.14}$$

(1.12) is a fourth order linear equation because the differential operators in (1.14) are the fourth order linear operator. After (1.12) is discretised on an image grid, the resulting equations are a linear system. Since the image is defined on a regular grid, the FFT solver can be employed very efficiently to solve (1.12). To use such a solver in a similar fashion as in Appendix A, we assume that the periodical boundary condition has been imposed on (1.12).

Another fourth order linear PDE we are concerned about has the form of

$$u(x) + \theta\Delta\left(\Delta u(x)\right) = g(x). \tag{1.15}$$

where θ and g have the same meaning as those in (1.9). If we use the definition (1.11), we have the following equation for the fourth order differential operator

$$\begin{aligned} \Delta\left(\Delta u\right) = {} & \partial_{x_1}\partial_{x_1}\partial_{x_1}\partial_{x_1}u + \cdots + \partial_{x_1}\partial_{x_1}\partial_{x_n}\partial_{x_n}u + \cdots \\ & + \partial_{x_n}\partial_{x_n}\partial_{x_1}\partial_{x_1}u + \cdots + \partial_{x_n}\partial_{x_n}\partial_{x_n}\partial_{x_n}u. \end{aligned} \tag{1.16}$$

If we compare the mixed fourth order partial derivatives in (1.16) against those in (1.14), we can notice that the order of the first order partial derivatives is slightly different. However, in accordance with the Clairaut's theorem, the mixed derivatives are equal if in each derivative we differentiate with respect to each variable the same number of times. In this sense, (1.12) is equivalent to (1.15) in a continuous setting.

Other PDEs that will appear in Chapter 2 are either a combination of (1.9), (1.12) and (1.15) or a part of them. Thereby, they can be well handled in the analogous manner to the second and fourth order linear PDEs present here. Later on in Chapter 2, we shall study how these PDEs can be discretised in an 2D image grid with the finite different method and what sort of numerical results they can lead to in terms of image denoising.

1.3.2 Hamilton-Jacobi equation

As mentioned previously, another category of PDEs is the nonlinear version. To this end, we present a Hamilton-Jacobi equation that belongs to the family of nonlinear PDEs and is also pervasive, especially in classical mechanics and geometrical optics. There exist modern mathematical tools that not only allow to analyse the properties of the solutions of Hamilton-Jacobi equations but also lay the foundation for many effective numerical algorithms, such as the fast sweeping method. These numerical methods are in general very efficient for solving the considered type of Hamilton-Jacobi equations in the viscosity sense.

A very famous example of a first order nonlinear PDE is the Eikonal equation, which has applications in geometric optics, wave propagation, and level set methods. It is the special case of Hamilton-Jacobi equations and has the form of

$$|\nabla u(x)| = v(x), \text{ for } x \in \Omega \subset \mathbb{R}^n, \tag{1.17}$$

which is subject to $u|_{\partial\Omega} = 0$. Here $v : \Omega \to (0, +\infty)$ is an input function with positive values and is closely related to the input image function f in image processing applications. ∇ is the gradient operator defined in (1.5) and $|\cdot|$ is the Euclidean norm. From physical point of view, the solution $u(x)$ is the shortest time required to travel from the boundary $\partial\Omega$ to x inside Ω, with $v(x)$ being the speed at x. In the special case when $v(x) = 1$, the solution reduces to the distance function from the boundary $\partial\Omega$ to x. (1.17) is said to be first order nonlinear because the Euclidean norm $|\nabla u|$ is a first order nonlinear differential operator. There is no way to express the discretised version of (1.17) in terms of a system of linear equations.

There are many computational tools that can seek a numerical solution to this nonlinear equation. For instance, an optimal control based approach which makes use of the Dijkstra's shortest path algorithm [4] was proposed in [5]. Label-correcting methods such as the Bellman–Ford algorithm [6, 7] were also used to solve the discretised Eikonal equation. Moreover, the fast marching method (FMM) [8, 9] and the fast sweeping method (FSM) [10, 11] are two very efficient approaches for this task. FMM uses an

upwind finite difference approximation. Its computational complexity is $O(Nlog(N))$ where N is the total number of grid points and the *log* term comes from the use of a heap (typically binary). FSM is based on a pre-defined sweep strategy, replacing the heap priority queue to find the next point to process, and thereby has the linear complexity of $O(N)$. FSM is faster than FMM for simple geometry problems. However, the situation may be reversed for complex geometry. In addition, hybrid methods [12, 13] have also been introduced that take advantage of FMM's efficiency and FSM's simplicity. For example, the heap cell method was developed to first decompose the computational domain into cells. Then FMM was performed on the cell-domain, while FSM was performed on the local gridpoint-domain that lies within that cell.

Solving the Eikonal equation (1.17) has two very important image processing applications in this thesis. The first one is the optical tomography image segmentation, which will be introduced in Chapter 4. Another one is surface reconstruction from point clouds, which will be detailed in Chapter 5. In Appendix B, we show the full implementation procedures of FSM on both 2D and 3D image grids.

1.3.3 Connections between variational and PDE-based methods

In general, PDE-based methods are closely related to variational methods. Minimisation of a variational functional normally results in a Euler-Lagrange equation that can be regarded as the steady-state of a suitable PDE [14]. On the other hand, certain PDEs can be also derived from the corresponding variational functionals. For instance, minimising the total variation energy functional $\int_\Omega |\nabla u| dx$ [15] gives rise to the Perona–Malik model [16], which is a PDE. Nevertheless, for several interesting types of PDEs, like the first order Hamilton-Jacobi equations, the corresponding PDEs are normally directly designed, without any known variational interpretation. To this end, PDE-based and variational methods have their own advantages when compared to each other. PDE-based methods are very easy to design and use, while variational approaches can easily integrate constraints imposed by problems and allow to employ powerful modern optimisation techniques such as primal dual [17, 18], fast iterative shrinkage-thresholding algorithm [19, 20], split Bregman [21–25], and augmented Lagrangian [26–28]. Recent advances [18, 20, 29] on how to automatically select parameters for different optimisation algorithms dramatically boost performance of variational methods, leading to further research interest in this field.

In summary, the basis of variational methods is the variational principle used in the calculus of variations, a field of mathematical analysis that deals with optimising functionals. These functionals are often expressed as integrals involving functions and their derivatives and thus variational methods involve solving PDEs. A PDE is an equation that involves function variables and their partial derivatives, which can be either linear or nonlinear.

1.4 Image processing applications

1.4.1 Image denoising

For image denoising, a functional normally has a data fidelity term and a regularisation term. The former allows the similarity between the recovered image and the original image, whilst the latter imposes some kind of smoothness/regularity on the recovered image. In the following, we use such a functional to model the image denoising problem.

Let us start with a more general image processing problem. Assume that a 2D grayscale image u can be considered as a real valued function, which is defined as

$$u : (\Omega \subset \mathbb{R}^2) \to \mathbb{R}. \tag{1.18}$$

In the following, we omit the function variable x for simplicity (e.g., we use a shorthand u for $u(x)$). Suppose that we have some image acquisition device, and we want the device to provide us with a perfect image u. However, due to imperfections and limitations in the acquisition process, we often get a corrupted version f of u. For example, corruptions can be noise, blur, etc. If we model such acquisition process mathematically, we have

$$f = Tu + \eta, \tag{1.19}$$

where η denotes a random noise component that follows a certain distribution. T is a forward operator mapping from u to f. Choices of T are various depending on different imaging tasks. For example, T is an identity in image denoising, T is a convolution blurring kernel in image deblurring, T is the Fourier transform in MRI image reconstruction, T is the Radon transform in computerised tomography (CT) image reconstruction, and T is a characteristic/indicator function in image inpainting, to name a few.

Now we need to solve u from (1.19) to get the clearer image. Note that this equation has a unique solution if it is a well-posed system. However, because the operator T in reality is most likely non-invertible (or even if it would be invertible, its inversion may be

ill-conditioned), (1.19) is ill-posed most time. In addition, it contains measurement noise and there may exist modelling errors, so solving the equation may be further complicated. In this case, some prior information is often added to the model, which in general is given by a certain regularity assumption on u. This procedure is the regularisation process introduced above. Specifically, the following functional is minimised in order to recover u from f.

$$E(u) = \frac{1}{2}\int_{\Omega} (Tu - f)^2 dx + \alpha \int_{\Omega} \mathcal{R}(u) dx, \tag{1.20}$$

where the first term on the right-hand side is the quadratic data fidelity or similarity term, which measures the distance between the data f and the reconstruction u. This fidelity term assumes that the random noise η is a Gaussian distribution. Other choices for this term based on different noise distribution are also available. We however only consider the Gaussian distribution in the thesis. The second term is the regulariser or regularisation term, which imposes extra regularity on u. These two terms are balanced by the positive parameter α, which controls the level of smoothness of resulting recovered image. Images acquired by medical devices usually contain a significant amount of noise which reduces the image quality, so (1.20) is adapted to merely image denoising as follows

$$E(u) = \frac{1}{2}\int_{\Omega} (u - f)^2 dx + \alpha \int_{\Omega} \mathcal{R}(u) dx. \tag{1.21}$$

The minimisation of (1.21) with a right selection of α usually leads to a desirable denoising result, which can be used for next stage image processing tasks, such as image segmentation. $\mathcal{R}(u)$ in the regularisation term should be designed appropriately for different type images, e.g., piecewise constant, piecewise smooth, texture, etc. In Chapter 2, we shall use (1.21) with different regularisation terms associated with first or/and second derivative to denoise images with different features.

1.4.2 Image segmentation

Image segmentation is to partition an image into multiple regions of similar characteristics such that computers can further understand and analyse the content within the image. To facilitate clinical studies of diseases, in this thesis we are particularly interested in segmenting images of three categories, that are red blood cell (RBC) images, histology images and optical coherence tomography (OCT) images. The clinical applications are as follows.

RBC image segmentation is to detect malaria parasite infected blood cells (iRBCs) and study anti-malarial drugs. Malaria, one of the most devastating infectious diseases around

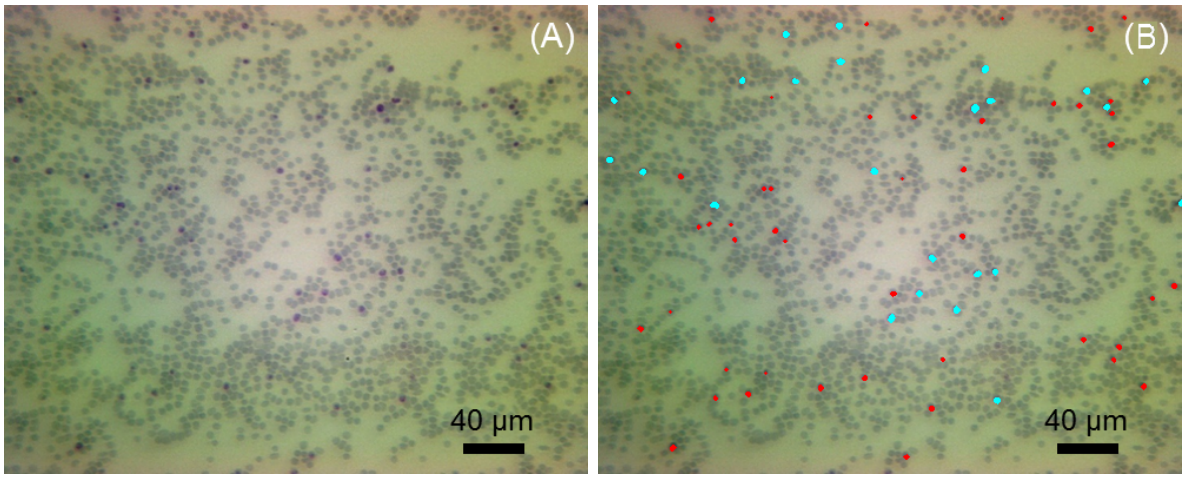


Figure 1.2: Automated classification of malaria parasite iRBCs by the developed image-based algorithm in Chapter 3. (A): Original image taken by objective 20X image cytometer. (B): The segmented iRBCs are classified into two groups by quantifying the area of them.

the globe, is caused by protozoan parasites of the genus Plasmodium. There are five major species of plasmodia that infect humans, out of which Plasmodium falciparum causes the majority of morbidity and mortality in Africa, followed by less lethal Plasmodium vivax infections across South East Asia. Altogether, they are infecting 200 million people and resulting in over half a million deaths every year [30]. Early diagnosis and treatment are required to avoid malaria-associated deaths [31]. Although anti-malarial drugs have been widely used to prevent and cure parasite-related diseases, parasites have acquired resistance to most of drugs both in Africa and South-East Asia, rendering new anti-malarial drug tests imperative. This is how image segmentation comes to play an important role. The idea is to segment the images formed using Giemsa-stained smears[1] on iRBCs, as shown in Figure 1.2 (A). Afterwards, we can detect and numerically counts malaria parasite iRBCs and classify all parasitic subpopulations by quantifying the area occupied by the parasites in iRBCs, as shown in Figure 1.2 (B). In this way, the efficacy of the anti-malarial drug can be tested by quantitatively comparing the iRBCs within the image before and after the drug is administrated.

Histology image segmentation is to detect microglial cells and study brain related diseases. Microglia are immune cells exclusive to the central nervous system and about 1.5 trillion of them reside in the brain and spinal cord [32]. They are essential to learning and memory [35, 36] and protect neurons from damage and they also mediate pain [33]. Microglial activation is a hallmark of chronic neuroin ammation, which plays

[1] A method designed primarily for the demonstration of parasites in malaria.

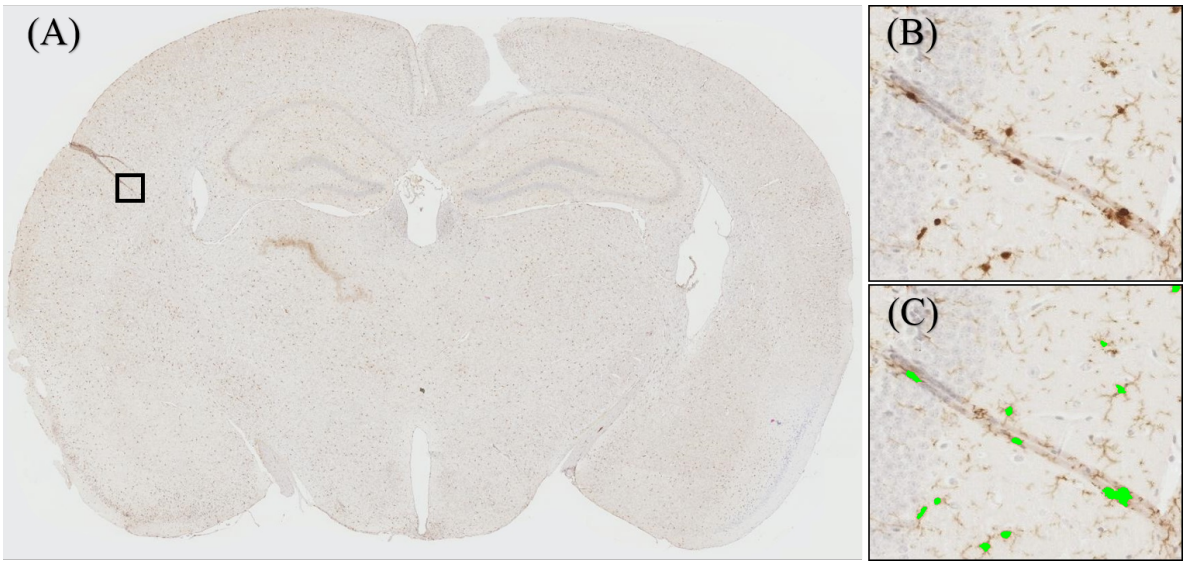


Figure 1.3: Segmentation of microglial cells from a histology image. (A): Whole brain slice of a healthy mouse; (B): Image region highlighted in the black rectangle in (A); (C) Segmentation results of the microglial cells in (B) using the method developed in Chapter 3.

an important role in a range of brain disorders, including stroke, multiple sclerosis, Parkinson's, Huntington's and Alzheimer's diseases [34–36] and other chronic conditions. Extracting the microglial cells is normally the first step to analyse microglial activation. One of the major challenges in quantitative microglial analysis from histology images is the development of automated microglial segmentation methods. Manual or semi-automated segmentation methods are extremely time-consuming and require user intervention [37, 38] with an element of subjectivity and inter-observer variability. Image analysis approaches that are commonly used for quantifying histology and that rely on thresholding, such as ImageJ [39], struggle with intensity inhomogeneity within histology images. Figure 1.3 shows an example of segmenting the microglial cells from a part of histology image derived from the whole brain slice of a healthy mouse.

OCT image segmentation is to delineate retinal layers and study optical nerve diseases. OCT is a powerful imaging modality used to image various aspects of biological tissues, such as structural information, blood flow, elastic parameters, change of polarization states and molecular content [40]. By using the low coherence interferometry, OCT can generate 2D/3D images from biological samples and provide high-resolution cross-sectional backscattering profiles. With a long period development of this technique, OCT has become a well-established modality for depth resolved imaging of eyes. Ophthalmology has drastically benefited from the inventions and improvements made to OCT systems.

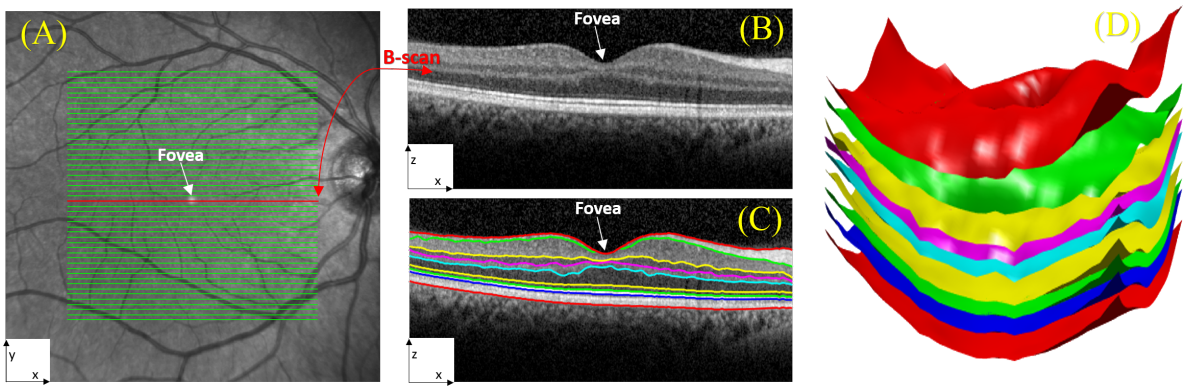


Figure 1.4: A en-face fundus image (A) overlaid with lines representing the locations of B-scans in a volumetric OCT image. The red line in (A) corresponds to the B-scan shown in (B). Nine target intra-retinal layer boundaries in 2D (C) and 3D (D) detected by the proposed method in Chapter 4.

In order to aid the diagnosis of pathologies by physicians, measurements are normally required to perform on the in-vivo imaging of retina in OCT images. One example of various measurements is to determine the thickness of retinal layers and segmentation thus has to be done before the measurement can be made. This segmentation task involves determining the location of nine retinal boundaries/surfaces in the OCT image. Once these retinal layers are delineated separately, spatial extent, statistics distribution and gradation of pixel intensities of the region between retinal layers can be calculated straightforwardly. Furthermore, the segmented retinal layers can be individually visualised, manipulated and analysed. Figure 1.4 shows an example of segmenting the retinal layers from the 2D/3D OCT images obtained from a healthy human in-vivo eye.

1.4.3 Surface reconstruction

We reconstruct 3D object surfaces from point clouds. A point cloud is a set of data points in some coordinate system. In a 3D coordinate system, these points are usually defined by x, y, and z coordinates, and often are intended to represent the external surface of an object. Point clouds may be created by 3D scanners. These devices measure a large number of points on an object's surface. As the output of 3D scanning processes, point clouds are used for many purposes, including to create 3D models for manufactured parts, metrology/quality inspection, and a multitude of visualization, animation, rendering and mass customization applications. While point clouds can be directly rendered and inspected, point clouds themselves are usually not directly usable in most 3D applications,

and therefore are usually converted to polygon mesh or triangle mesh through a process commonly referred to as surface reconstruction.

A major difference between surface reconstruction from point clouds and image segmentation is that there is no initial 2D or 3D image for the segmentation. All we know is the locations of the scattered 2D or 3D points. Since there exist effective and efficient algorithms for image segmentation using variational methods, we transform the surface reconstruction task to the variational image segmentation task, partitioning the computational domain into two regions, interior and exterior. To do so, we will need to create an initial image and an edge indicator directly from the given point cloud. The initial image should provide a good initialisation for the segmentation task, and the edge indicator function should provide more information about the precise location of the image boundary. Statistically speaking, the initial image function gives the likelihood of a grid point being inside or outside, and the edge indicator function gives the likelihood of a grid point on the image boundary. The initial image and edge indicator can be then effectively integrated to an energy functional capable of reconstructing the volume, as shown in Figure 1.5. The point cloud data used in the thesis are downloaded from https://graphics.stanford.edu/data/3Dscanrep/.

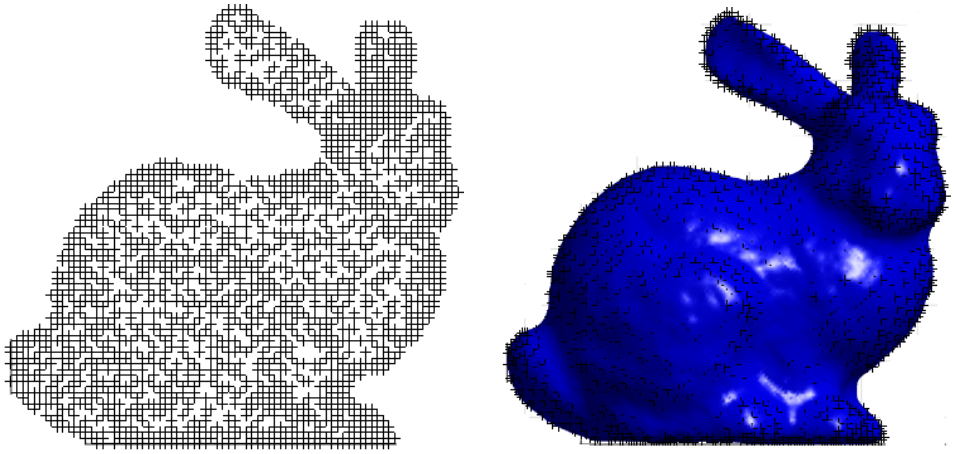


Figure 1.5: Surface reconstruction from a point cloud. Left: Point cloud of the Stanford Bunny; Right: Surface reconstruction result using the proposed method in Chapter 5.

1.5 Organisation of the thesis

The remaining chapters of this thesis are organised as follows:

In Chapter 2, we study image denoising. We present some representative first and second order variational models for image denoising, provide detailed descretisation of

the partial derivatives in the resulting PDEs, and perform numerical implementation of these models based on the FFT and the split Bregman algorithm. We also demonstrate the advantages and disadvantages of these variational models through extensive image denoising experiments.

In Chapter 3, we study image segmentation for two types images. First, we developed a novel image analysis framework for an image-based cytometer. Such a framework can detect, count and analyse malaria parasite iRBCs from Giemsa-stained smears. The image analysis framework includes the variational total generalised variation, adaptive thresholding and support vector machine classification. Second, we introduce an automated image approach framework capable of efficiently segmenting microglial cells from histology images. We show the complete implementation of the Mumford-Shah total variation using the split Bregman algorithm for denoising and segmentation of microglial cells. We test the accuracy of the developed methods using the manual segmentation results and large datasets.

In Chapter 4, we study OCT image segmentation. We introduce a novel and accurate geodesic distance method for segmentation of both healthy and pathological OCT images in either 2D or 3D. The results of the proposed method are evaluated using manual segmentation. We also compare the proposed approach with the parametric active contour algorithm and graph-based approaches through extensive numerical results.

In Chapter 5, we study surface reconstruction from point clouds. We introduce a novel variational level set method for reconstructing an accurate implicit surface from a set of unorganised points. Implementation details of the variational model are given, including the discretisiation of the 3D differential operators and the use of the gradient descent flow. Extensive experiments are conducted to compare the proposed method with the state-of-the-art surface reconstruction approaches.

In Chapter 6, we conclude the thesis and give some comments on future work.

In Appendix A, we present the FFT solver for a second order linear PDE.

In Appendix B, we present the numerical implementation of fast sweeping in both 2D and 3D.

Chapter 2

Image pre-processing

In this chapter, we introduce image denoising as an image pre-processing step for the segmentation problems described in the later chapters. Specifically, we focus on the variational method, a powerful approach that has been widely used in image processing community. We briefly review some representative first and second order variational models for this task. The investigated denoising approaches include the quadratic Tikhonov (QT), total variation (TV), absolute Laplacian (AL), bounded Hessian (BH), hybridised TV and Laplacian (TVL), hybridised TV and bounded Hessian (TVBH), total generalised variation (TGV), and absolute curvature (AC) models. The differential operators appeared in these models and their corresponding Euler-Lagrange equations are then discretised via the finite difference scheme, followed by the fast numerical implementation of these denoising models based on the split Bregman algorithm. We end the chapter by drawing the conclusions about the advantages and disadvantages of these models through extensive comparative experiments. We denote $\int_\Omega$ as $\int_\Omega dx$ in the following sections.

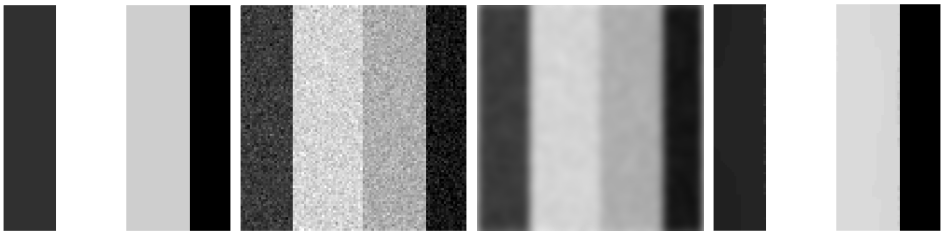
2.1 First order variational models

In the last three decades or so, numerous variational models [41, 3, 42, 43] have been applied to image processing. One of the most popular variational models for image denoising is the quadratic Tikhonov (QT) model, and its energy functional is given by

$$E(u) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega |\nabla u|^2, \tag{2.1}$$

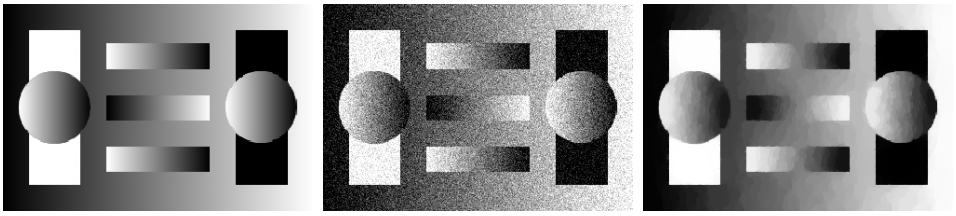
where $f : (\Omega \subset \mathbb{R}^2) \to \mathbb{R}$ is the input noisy image and u is the output denoised image. Minimising the model through the variational method gives a linearly smoothed image u

in which the smoothing strength for each pixel is isotropic. Therefore, the underlying structures like edges are blurred, even though the model eliminates noise in the given data. See Figure 2.1.

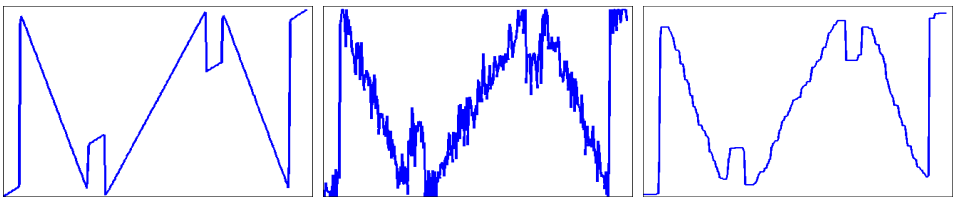


(a) Clean image (b) Noisy image (c) QT (d) TV

Figure 2.1: Illustration of the isotropically smoothed denoising result using QT and the edge preservation denoising result using TV.



(a) Clean image (b) Noisy image (c) TV



(d) Middle slice of (a) (e) Middle slice of (b) (f) Middle slice of (c)

Figure 2.2: Illustration of the staircase effect in TV denoising.

This observation motivates a new class of regularisers, which aims to remove noise and smooth the image in homogeneous areas, while preserve edges and boundaries of objects in the image. In their pioneering work, Rudin Osher and Fatemi [15] proposed to use the total variation (TV) model, and the energy functional of the model is as follows

$$E(u) = \frac{1}{2}\int_{\Omega}(u-f)^2 + \alpha\int_{\Omega}|\nabla u|. \tag{2.2}$$

This first order TV model is effective in preserving object edges while removing noise, as shown in Figure 2.1 (d). A desirable property of image denoising algorithms is to restore images without losing important features (i.e. edges, corners, contrast, etc.). However, the TV model has several undesirable side effects, one of which is the staircase effect, that

is, the restored image appears jagged, see Figure 2.2. Higher order variational models have been extensively proposed to eliminate the side effect.

2.2 Higher order variational models

These higher order variational models studied in this thesis are briefly reviewed below. The simplest higher order models involve only second order derivatives. For instance, a model that uses the second order absolute Laplace regulariser, named the AL model in this thesis, was proposed in [44]

$$E(u) = \frac{1}{2}\int_{\Omega}(u-f)^2 + \alpha\int_{\Omega}|\Delta u|, \tag{2.3}$$

where the Laplacian is defined as $\Delta u = \partial_x\partial_x u + \partial_y\partial_y u$, with ∂_x and ∂_y being the first order partial derivatives along x and y directions, respectively.

A model that uses the second order bounded Hessian regulariser, named the BH model in this thesis, was proposed by Lysaker et al [45], Scherzer et al [46, 47], Lai at al [48], and Bergounioux et al [49]. The model minimises the following energy functional

$$E(u) = \frac{1}{2}\int_{\Omega}(u-f)^2 + \alpha\int_{\Omega}\left|\nabla^2 u\right|, \tag{2.4}$$

where the symmetric Hessian matrix of u (i.e. $\nabla^2 u$) in the image domain Ω is given as follows

$$\nabla^2 u = \begin{pmatrix} \partial_x\partial_x u & \partial_y\partial_x u \\ \partial_x\partial_y u & \partial_y\partial_y u \end{pmatrix}. \tag{2.5}$$

A model that combines the TV and Laplacian regulariser, named the TVL model in this thesis, was proposed in [50–52]

$$E(u) = \frac{1}{2}\int_{\Omega}(u-f)^2 + \alpha\int_{\Omega}|\nabla u| + \beta\int_{\Omega}|\Delta u|. \tag{2.6}$$

A model that combines the TV and bounded Hessian regulariser, named the TVBH model, was proposed in [22, 23]

$$E(u) = \frac{1}{2}\int_{\Omega}(u-f)^2 + \alpha\int_{\Omega}|\nabla u| + \beta\int_{\Omega}\left|\nabla^2 u\right|. \tag{2.7}$$

A model named the total generalised variation (TGV) was proposed in [53] for image denosing. The main feature of TGV is that it can deal with different image characteristics (e.g. piecewise constant, piecewise affine, piecewise quadratic, etc.). The second order

TGV model is considered here and is of the following form

$$E(u,p) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega |\nabla u - p| + \beta \int_\Omega |\varepsilon(p)|, \tag{2.8}$$

where $\varepsilon(p)$ is the symmetrised derivative and defined as

$$\varepsilon(p) = \begin{pmatrix} \partial_x p_1 & \frac{\partial_y p_1 + \partial_x p_2}{2} \\ \frac{\partial_y p_1 + \partial_x p_2}{2} & \partial_y p_2 \end{pmatrix}. \tag{2.9}$$

The minimum of (2.8) is taken over all gradients of the deformation field $p = (p_1, p_2)$ on image space Ω.

A model that uses the absolute curvature of an image as a regulariser, named the AC model in this thesis, was introduced in [54–57] for image denoising. The AC model is given as

$$E(u) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega \left| div\left(\frac{\nabla u}{|\nabla u|}\right)\right|. \tag{2.10}$$

In fact, the curvature-based models were first introduced by Nitzberg and Mumford [58] in early 1990s and then extended for image inpainting [59–61], segmentation with depth [62], construction of illusory shapes [63] etc.

Despite of the effectiveness of these higher order models in removing the staircase effect, it is often a challenge to minimise the corresponding functionals. Technically, the Euler-Lagrange equations of the regularisers are fourth order nonlinear partial differential equations (PDEs), which are very difficult to discretise to solve computationally. In this thesis, the split Bregman algorithm [21, 64, 65] is applied to transform the energy minimisation problem of a given model into several subproblems. These subproblems are then efficiently solved by the fast Fourier transform (FFT) [25, 66, 26, 27, 56], analytical soft thresholding equation [67] and projection formula [27, 56] without any iteration.

2.3 Discretisation of differential operators

In order to solve the variational models above using the efficient split Bregman method [21, 68, 69], the first, second and fourth order derivative operators are discretised using the finite difference scheme. These differential operators are respectively ∂_x^+, ∂_x^-, ∂_y^+, ∂_y^-, $\partial_x^-\partial_x^+$, $\partial_x^+\partial_x^-$, $\partial_y^-\partial_y^+$, $\partial_y^+\partial_y^-$, $\partial_y^+\partial_x^+$, $\partial_x^+\partial_y^+$, $\partial_y^-\partial_x^-$, $\partial_x^-\partial_y^-$, $\partial_x^+\partial_y^-$, $\partial_y^+\partial_x^-$, $\partial_x^+\partial_x^-\partial_x^-\partial_x^+$, $\partial_y^+\partial_y^-\partial_y^-\partial_y^+$, $\partial_y^-\partial_x^-\partial_y^+\partial_x^+$ and $\partial_x^-\partial_y^-\partial_x^+\partial_y^+$. The gradient, Hessian matrix (2.5) and sym-

metrised derivative (2.9) are discretised as follows respectively

$$\nabla u = \left(\partial_x^+ u \;\; \partial_y^+ u\right), \tag{2.11}$$

$$\nabla^2 u = \begin{pmatrix} \partial_x^- \partial_x^+ u & \partial_y^+ \partial_x^+ u \\ \partial_x^+ \partial_y^+ u & \partial_y^- \partial_y^+ u \end{pmatrix}, \tag{2.12}$$

$$\varepsilon(\boldsymbol{p}) = \begin{pmatrix} \partial_x^- p_1 & \frac{\partial_y^- p_1 + \partial_x^- p_2}{2} \\ \frac{\partial_y^- p_1 + \partial_x^- p_2}{2} & \partial_y^- p_2 \end{pmatrix}. \tag{2.13}$$

Theoretically, $\partial_x \partial_y u$ should be equal to $\partial_y \partial_x u$ in the Hessian matrix (2.5), so it is necessary to define its discrete form such that $\partial_x^+ \partial_y^+ u = \partial_y^+ \partial_x^+ u$ in (2.12). The issue of boundary conditions also should be addressed, as it describes how discrete derivatives are defined on boundaries. Unlike the Neumann boundary condition used in [70–72], we apply the periodic boundary condition. By choosing such boundary condition, FFT can be conveniently utilised for the split Bregman algorithm such that the computational speed can be improved.

The first order forward and backward difference schemes are first given. Let $\Omega \to \mathbb{R}^{M \times N}$ denote a 2D greyscale image space with size $M \times N$. The coordinates x and y are oriented along columns and rows respectively. So the first order forward differences of u at point (i, j) along x and y directions are respectively

$$\partial_x^+ u_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } 1 \le \mathrm{i} \le \mathrm{M},\ 1 \le \mathrm{j} < \mathrm{N} \\ u_{i,1} - u_{i,j} & \text{if } 1 \le \mathrm{i} \le \mathrm{M},\ \mathrm{j} = \mathrm{N} \end{cases}, \tag{2.14}$$

$$\partial_y^+ u_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } 1 \le \mathrm{i} < \mathrm{M},\ 1 \le \mathrm{j} \le \mathrm{N} \\ u_{1,j} - u_{i,j} & \text{if } \mathrm{i} = \mathrm{M},\ 1 \le \mathrm{j} \le \mathrm{N} \end{cases}. \tag{2.15}$$

The first order backward differences are respectively

$$\partial_x^- u_{i,j} = \begin{cases} u_{i,j} - u_{i,j-1} & \text{if } 1 \le \mathrm{i} \le \mathrm{M},\ 1 < \mathrm{j} \le \mathrm{N} \\ u_{i,j} - u_{i,N} & \text{if } 1 \le \mathrm{i} \le \mathrm{M},\ \mathrm{j} = 1 \end{cases}, \tag{2.16}$$

$$\partial_y^- u_{i,j} = \begin{cases} u_{i,j} - u_{i-1,j} & \text{if } 1 < \mathrm{i} \le \mathrm{M},\ 1 \le \mathrm{j} \le \mathrm{N} \\ u_{i,j} - u_{M,j} & \text{if } \mathrm{i} = 1\ ,\ 1 \le \mathrm{j} \le \mathrm{N} \end{cases}. \tag{2.17}$$

For every $\boldsymbol{p} = (p_1 \;\; p_2) \in (\mathbb{R}^{M\times N})^2$ and $u \in \mathbb{R}^{M\times N}$, the discrete version of the first order adjoint divergence operator satisfies

$$\sum_{\substack{1\le i\le M\\ 1\le j\le N}} -div\left(\boldsymbol{p}_{i,j}\right)\cdot u_{i,j} = \sum_{\substack{1\le i\le M\\ 1\le j\le N}} \boldsymbol{p}_{i,j}\cdot \nabla u_{i,j}.$$

Therefore, according to the definition of discrete gradient operator, it is easy to check the discrete divergence as

$$div\left(\boldsymbol{p}_{i,j}\right) = \partial_x^- p_{1i,j} + \partial_y^- p_{2i,j}. \tag{2.18}$$

The discrete second order derivatives $\partial_x^-\partial_x^+u$, $\partial_x^+\partial_x^-u$, $\partial_y^-\partial_y^+u$, $\partial_y^+\partial_y^-u$, $\partial_y^+\partial_x^+u$, $\partial_x^+\partial_y^+u$, $\partial_y^-\partial_x^-$ and $\partial_x^-\partial_y^-$ at point (i,j) can be written down as follows respectively. These operators are just the corresponding compositions of the discrete first order derivative.

$$\partial_x^+\partial_x^- u_{i,j} = \partial_x^-\partial_x^+ u_{i,j} = \begin{cases} u_{i,N} - 2u_{i,j} + u_{i,j+1} & \text{if } 1\le \mathrm{i}\le \mathrm{M},\ \mathrm{j}=1\\ u_{i,j-1} - 2u_{i,j} + u_{i,j+1} & \text{if } 1\le \mathrm{i}\le \mathrm{M},\ 1<\mathrm{j}<\mathrm{N}\\ u_{i,j-1} - 2u_{i,j} + u_{i,1} & \text{if } 1\le \mathrm{i}\le \mathrm{M},\ \mathrm{j}=\mathrm{N}\end{cases}, \tag{2.19}$$

$$\partial_y^+\partial_y^- u_{i,j} = \partial_y^-\partial_y^+ u_{i,j} = \begin{cases} u_{M,j} - 2u_{i,j} + u_{i+1,j} & \text{if } \mathrm{i}=1,\ 1\le \mathrm{j}\le \mathrm{N}\\ u_{i-1,j} - 2u_{i,j} + u_{i+1,j} & \text{if } 1<\mathrm{i}<\mathrm{M},\ 1\le \mathrm{j}\le \mathrm{N}\\ u_{i-1,j} - 2u_{i,j} + u_{1,j} & \text{if } \mathrm{i}=\mathrm{M},\ 1\le \mathrm{j}\le \mathrm{N}\end{cases}, \tag{2.20}$$

$$\partial_x^+\partial_y^+ u_{i,j} = \partial_y^+\partial_x^+ u_{i,j} = \begin{cases} u_{i,j} - u_{i+1,j} - u_{i,j+1} + u_{i+1,j+1} & \text{if } 1\le \mathrm{i}<\mathrm{M},\ 1\le \mathrm{j}<\mathrm{N}\\ u_{i,j} - u_{1,j} - u_{i,j+1} + u_{1,j+1} & \text{if } \mathrm{i}=\mathrm{M},\ 1\le \mathrm{j}<\mathrm{N}\\ u_{i,j} - u_{i+1,j} - u_{i,1} + u_{i+1,1} & \text{if } 1\le \mathrm{i}<\mathrm{M},\ \mathrm{j}=\mathrm{N}\\ u_{i,j} - u_{1,j} - u_{i,1} + u_{1,1} & \text{if } \mathrm{i}=\mathrm{M},\ \mathrm{j}=\mathrm{N}\end{cases}, \tag{2.21}$$

$$\partial_x^-\partial_y^- u_{i,j} = \partial_y^-\partial_x^- u_{i,j} = \begin{cases} u_{i,j} - u_{i,N} - u_{M,j} + u_{M,N} & \text{if } \mathrm{i}=1,\ \mathrm{j}=1\\ u_{i,j} - u_{i,j-1} - u_{M,j} + u_{M,j-1} & \text{if } \mathrm{i}=1,\ 1<\mathrm{j}\le \mathrm{N}\\ u_{i,j} - u_{i,N} - u_{i-1,j} + u_{i-1,N} & \text{if } 1<\mathrm{i}\le \mathrm{M},\ \mathrm{j}=1\\ u_{i,j} - u_{i,j-1} - u_{i-1,j} + u_{i-1,j-1} & \text{if } 1<\mathrm{i}\le \mathrm{M},\ 1<\mathrm{j}\le \mathrm{N}\end{cases}. \tag{2.22}$$

Note that the constraint $\partial_x\partial_y u = \partial_y\partial_x u$ in the Hessian matrix (2.5) is satisfied as now the discrete form $\partial_x^+\partial_y^+u_{i,j} = \partial_y^+\partial_x^+u_{i,j}$ in (2.21).

Based on (2.19) and (2.20), the definition of discrete Laplace operator is given as

$$\Delta u_{i,j} = div(\nabla u_{i,j}) = \partial_x^-\partial_x^+u_{i,j} + \partial_y^-\partial_y^+u_{i,j}. \tag{2.23}$$

In addition, we need the discrete second order divergence operator that also has adjointness property: for every $\boldsymbol{q} = \begin{pmatrix} q_1 & q_2 \\ q_3 & q_4 \end{pmatrix} \in (\mathbb{R}^{M\times N})^4$ and $u \in \mathbb{R}^{M\times N}$, we have

$$\sum\nolimits_{\substack{1\le i\le M\\ 1\le j\le N}} div^2(\boldsymbol{q}_{i,j}) \cdot u_{i,j} = \sum\nolimits_{\substack{1\le i\le M\\ 1\le j\le N}} \boldsymbol{q}_{i,j} \cdot \nabla^2 u_{i,j}.$$

Then the discrete second order divergence reads

$$div^2\left(\boldsymbol{q}_{i,j}\right) = \partial_x^+\partial_x^- q_{1i,j} + \partial_y^-\partial_x^- q_{2i,j} + \partial_x^-\partial_y^- q_{3i,j} + \partial_y^+\partial_y^- q_{4i,j}. \tag{2.24}$$

In order to implement these higher order variational models, we give the discrete definitions of another two second order derivatives

$$\partial_x^+\partial_y^- u_{i,j} = \begin{cases} u_{i.j+1} - u_{i,j} - u_{M,j+1} + u_{M,j} & \text{if i} = 1,\ 1 \le \text{j} < \text{N} \\ u_{i.1} - u_{i,j} - u_{M,1} + u_{M,j} & \text{if i} = 1,\ \text{j} = \text{N} \\ u_{i,j+1} - u_{i,j} - u_{i-1,j+1} + u_{i-1,j} & \text{if } 1 < \text{i} \le \text{M},\ 1 \le \text{j} < \text{N} \\ u_{i,1} - u_{i,j} - u_{i-1,1} + u_{i-1,j} & \text{if } 1 < \text{i} \le \text{M},\ \text{j} = \text{N} \end{cases}, \tag{2.25}$$

$$\partial_y^+\partial_x^- u_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} - \ u_{i,N} + u_{i,N} & \text{if } 1 \le \text{i} < \text{M},\ \text{j} = 1 \\ u_{1,j} - u_{i,j} - u_{1,N} + u_{i,N} & \text{if i} = \text{M},\ \text{j} = 1 \\ u_{i+1,j} - u_{i,j} - u_{i+1,j-1} + u_{i,j-1} & \text{if } 1 \le \text{i} < \text{M},\ 1 < \text{j} \le \text{N} \\ u_{1,j} - u_{i,j} - u_{1,j-1} + u_{i,j-1} & \text{if i} = \text{M},\ 1 < \text{j} \le \text{N} \end{cases}, \tag{2.26}$$

and the discrete fourth order derivatives

$$\partial_x^-\partial_x^+\partial_x^-\partial_x^+ u_{i,j} = \partial_x^+\partial_x^-\partial_x^-\partial_x^+ u_{i,j} = 6u_{i,j} + u_{i,j+2} + u_{i,j-2} - 4u_{i,j+1} - 4u_{i,j-1}, \tag{2.27}$$

$$\partial_y^-\partial_y^+\partial_y^-\partial_y^+ u_{i,j} = \partial_y^+\partial_y^-\partial_y^-\partial_y^+ u_{i,j} = 6u_{i,j} + u_{i+2,j} + u_{i-2,j} - 4u_{i+1,j} - 4u_{i-1,j}, \tag{2.28}$$

$$\begin{aligned} \partial_x^-\partial_x^+\partial_y^-\partial_y^+ u_{i,j} &= \partial_y^-\partial_y^+\partial_x^-\partial_x^+ u_{i,j} = \partial_x^-\partial_y^-\partial_x^+\partial_y^+ u_{i,j} = \partial_y^-\partial_x^-\partial_y^+\partial_x^+ u_{i,j} \\ &= 4u_{i,j} + u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1} \\ &\quad - 2\left(u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j}\right) \end{aligned}. \tag{2.29}$$

For simplicity, the finite differences on the boundaries for the discrete fourth order derivatives are not listed. One can easily compose the already-defined second order derivatives in (2.19)-(2.22) for the boundary conditions of the fourth order derivatives in (2.27)-(2.29). Based on (2.27)-(2.29), we can obtain following two discrete fourth order

differential operators

$$div^2\left(\nabla^2 u_{i,j}\right) = \partial_x^+\partial_x^-\partial_x^-\partial_x^+ u_{i,j} + \partial_y^-\partial_x^-\partial_y^+\partial_x^+ u_{i,j} + \partial_x^-\partial_y^-\partial_x^+\partial_y^+ u_{i,j} + \partial_y^+\partial_y^-\partial_y^-\partial_y^+ u_{i,j}, \tag{2.30}$$

$$\Delta(\Delta u_{i,j}) = \partial_x^-\partial_x^+\partial_x^-\partial_x^+ u_{i,j} + \partial_x^-\partial_x^+\partial_y^-\partial_y^+ u_{i,j} + \partial_y^-\partial_y^+\partial_x^-\partial_x^+ u_{i,j} + \partial_y^-\partial_y^+\partial_y^-\partial_y^+ u_{i,j}. \tag{2.31}$$

Note that $div^2\left(\nabla^2 u_{i,j}\right) = \Delta(\Delta u_{i,j})$. Figure 2.3 describes the behaviour of all the above discrete differential operators. Having defined all necessary discrete quantities, the numerical optimisation of the variational models in sections 4.1 and 2.2 can be implemented in the next.

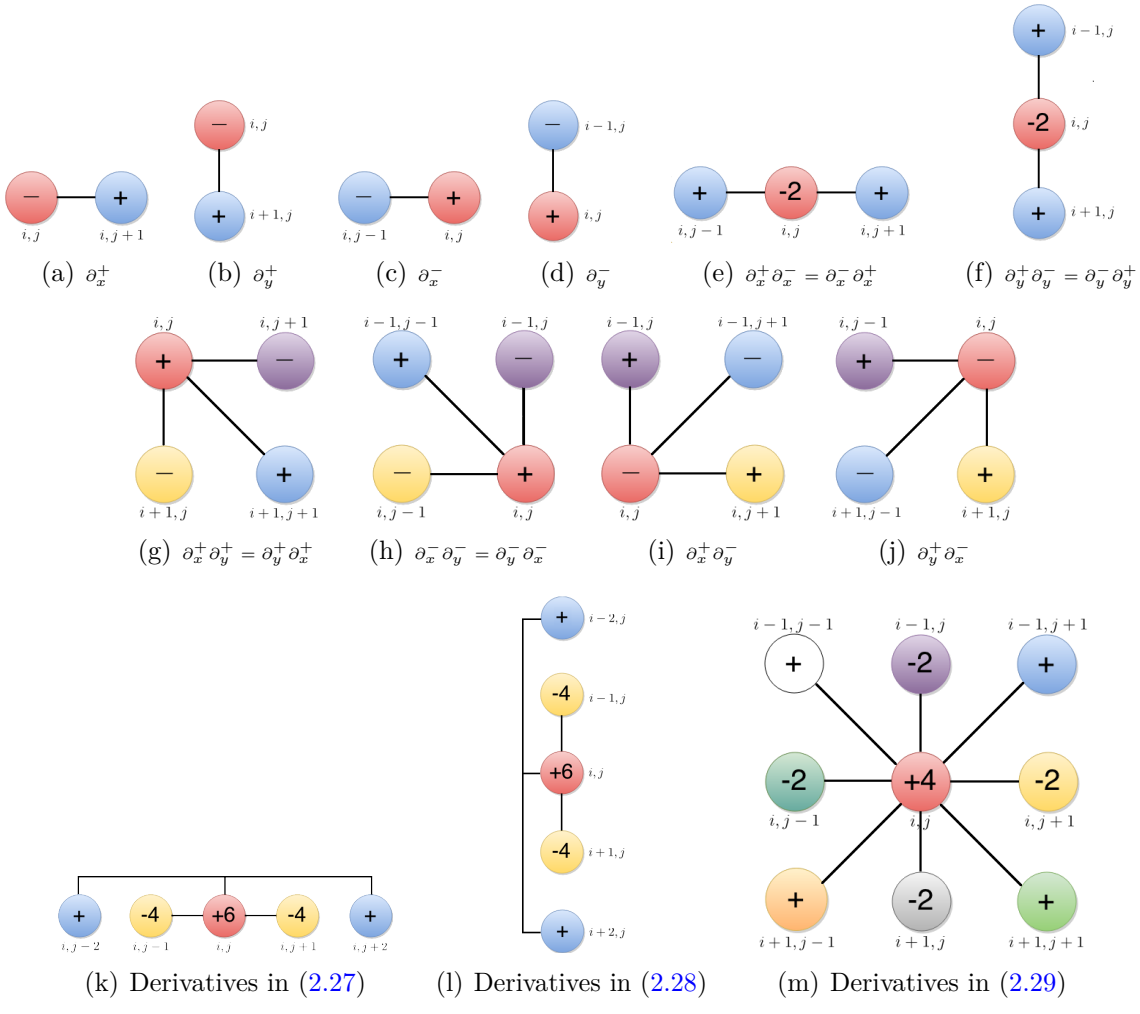


Figure 2.3: Graph illustration of the discrete first, second and fourth order derivative approximations.

2.4 Fast numerical implementation

In this section, we shall detail how to apply the fast split Bregman based numerical algorithms to the image denoising models introduced above. Since the QT model can be optimised directly, its implementation will be not given here. However, its solution process is similar to solving the subproblem with respect to u in the TV model.

2.4.1 Split Bregman for total variation (TV)

First, we introduce an auxiliary splitting vector variable $\boldsymbol{w} = (w_1\ w_2) \in (\mathbb{R}^{M\times N})^2$, a Bregman iterative parameter $\boldsymbol{b} = (b_1\ b_2) \in (\mathbb{R}^{M\times N})^2$ and a positive penalty parameter θ, transforming the functional (2.2) into the following form

$$E(u, \boldsymbol{w}; \boldsymbol{b}) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega |\boldsymbol{w}| + \frac{\theta}{2}\int_\Omega |\boldsymbol{w} - \nabla u - \boldsymbol{b}|^2, \tag{2.32}$$

In order to find the minimiser of the functional (2.2), we use an alternating optimisation method to search all the saddle points of the functional (2.32). To do so, we first fix the variable $\boldsymbol{w}$, deriving the Euler-Lagrange equation with respect to u of (2.32)

$$u - \theta div(\nabla u) = f - \theta div(\boldsymbol{w} - \boldsymbol{b}), \tag{2.33}$$

which is a linear Possion equation, for which there are many existing efficient solvers. In the next, we will see how the FFT solver can be applied to the equation. A detailed mathematical derivation can be found in Appendix A. We note that the FFT derivation for a fourth order linear PDE can be done in a similar fashion.

According to the definition of the discrete gradient (2.11), divergence (2.18) and Laplacian (2.23) in section 2.3, the discretisation form of (2.33) can be written as

$$u_{i,j} - \theta\left(\partial_x^- \partial_x^+ u_{i,j} + \partial_y^- \partial_y^+ u_{i,j}\right) = G_{i,j}, \tag{2.34}$$

with $G_{i,j} = f_{i,j} - \theta\left(\partial_x^-(w_{1i,j} - b_{1i,j}) + \partial_y^-(w_{2i,j} - b_{2i,j})\right)$. As the periodic boundary condition has been imposed on the discrete derivatives in section 2.3, the discrete Fourier transform can be directly applied to the both sides of the equation (2.34)

$$\mathcal{F}\left(u_{i,j} - \theta\left(\partial_x^- \partial_x^+ u_{i,j} + \partial_y^- \partial_y^+ u_{i,j}\right)\right) = \mathcal{F}(G_{i,j}),$$

where $\mathcal{F}$ denotes the discrete Fourier transform. For the discrete frequencies r and s, we have

$$\underbrace{\left(1-2\theta\left(\cos\frac{2\pi s}{N}+\cos\frac{2\pi r}{M}-2\right)\right)}_{\kappa}\mathcal{F}\left(u_{i,j}\right)=\mathcal{F}\left(G_{i,j}\right). \tag{2.35}$$

Here $i\in[1,M]$ and $j\in[1,N]$ are the indexes in the discrete time domain. $r\in[0,M)$ and $s\in[0,N)$ are the frequencies in the discrete frequency domain. Appendix A shows how the coefficient κ can be derived by using the FFT (2.35) provides us with a closed-form solution of u as

$$u_{i,j}=\Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(G_{i,j}\right)}{\kappa}\right)\right), \tag{2.36}$$

where $\mathcal{F}^{-1}$ denotes the discrete inverse Fourier transform. $\Re$ is the real part of a complex number. "—" stands for pointwise division of matrices, and κ is defined in (2.35).

After u is solved, we fix u for the vector $\boldsymbol{w}$ and obtain its Euler-Lagrange equation as follows

$$\alpha\frac{\boldsymbol{w}}{|\boldsymbol{w}|}+\theta\left(\boldsymbol{w}-\nabla u-\boldsymbol{b}\right)=0,$$

which can be solved component-wisely through the following discrete 2D analytical generalised soft thresholding equation with the convention that $0/0=0$

$$\boldsymbol{w}_{i,j}=\max\left(\left|\nabla u_{i,j}+\boldsymbol{b}_{i,j}\right|-\frac{\alpha}{\theta},0\right)\frac{\nabla u_{i,j}+\boldsymbol{b}_{i,j}}{\left|\nabla u_{i,j}+\boldsymbol{b}_{i,j}\right|}. \tag{2.37}$$

Finally, we update the Bregman iterative parameter $\boldsymbol{b}$, which is given in the step 6 in **Algorithm** 2.4.1. The overall algorithm for (2.2) reads as follows

Algorithm 2.4.1: Split Bregman algorithm for TV model (2.2)

1: ***function*** $\mathrm{TV}_{Denoising}(f)$
2: Initialisation: Set $(\boldsymbol{w};\boldsymbol{b})=0$, $(\alpha,\theta)>0$
3: ***repeat***
4: Compute u according to (2.36)
5: Compute $\boldsymbol{w}$ according to (2.37)
6: Update Bregman iterative parameter $\boldsymbol{b}\leftarrow\boldsymbol{b}+\nabla u-\boldsymbol{w}$
7: ***until*** some stopping criterion is satisfied[1]
8: ***return*** u
9: ***end function***

[1]In this thesis, we use $|E^k-E^{k-1}|/E^k\leq\epsilon$ as the stopping criterion, where E is the model energy, k is iteration number and threshold ϵ is fixed to $\epsilon=10^{-5}$. Alternatively, one can identify the convergence of function u by checking if the value of u changes below a threshold value between two consecutive iterations.

2.4.2 Split Bregman for absolute Laplacian (AL)

In order to minimise the AL model (2.3), an auxiliary splitting scalar variable $w \in \mathbb{R}^{M\times N}$, a Bregman iterative parameter $b \in \mathbb{R}^{M\times N}$ and a penalty positive parameter θ are introduced, transforming (2.3) into

$$E(u,w;b) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha\int_\Omega |w| + \frac{\theta}{2}\int_\Omega (w-\Delta u-b)^2.$$

Applying the alternating optimisation technique, we first fix w for u and obtain the following Euler-equation

$$u + \theta\Delta(\Delta u) = f + \theta\Delta(w-b). \tag{2.38}$$

Referring to the definitions of (2.23) and (2.31) in section 2.3, (2.38) is discretised as

$$u_{i,j} + \theta\left(\partial_x^-\partial_x^+\partial_x^-\partial_x^+ u_{i,j} + \partial_x^-\partial_x^+\partial_y^-\partial_y^+ u_{i,j} + \partial_y^-\partial_y^+\partial_x^-\partial_x^+ u_{i,j} + \partial_y^-\partial_y^+\partial_y^-\partial_y^+ u_{i,j}\right) = G_{i,j}, \tag{2.39}$$

with $G_{i,j} = f_{i,j} + \theta\left(\partial_x^-\partial_x^+(w_{i,j}-b_{i,j}) + \partial_y^-\partial_y^+(w_{i,j}-b_{i,j})\right)$. Applying the discrete Fourier transform to the both sides of the equation (2.39) gives

$$\mathcal{F}\left(u_{i,j} + \theta\left(\partial_x^-\partial_x^+\partial_x^-\partial_x^+ u_{i,j} + \partial_x^-\partial_x^+\partial_y^-\partial_y^+ u_{i,j} + \partial_y^-\partial_y^+\partial_x^-\partial_x^+ u_{i,j} + \partial_y^-\partial_y^+\partial_y^-\partial_y^+ u_{i,j}\right)\right) = \mathcal{F}(G_{i,j}). \tag{2.40}$$

For the discrete frequencies r and s, we have the following equivalent of the equation (2.40)

$$\underbrace{\left(1 + 4\theta\left(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2\right)^2\right)}_{\chi}\mathcal{F}(u_{i,j}) = \mathcal{F}(G_{i,j}). \tag{2.41}$$

(2.40) provides us with a closed-form solution of u as

$$u_{i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}(G_{i,j})}{\chi}\right)\right). \tag{2.42}$$

Next fixing the variable u to derive the following Euler-equation with respect to the scalar variable w

$$\alpha\frac{w}{|w|} + \theta(w-\Delta u-b) = 0,$$

which can be solved component-wisely through the following discrete 1D analytical generalised soft thresholding equation with the convention that $0/0 = 0$

$$w_{i,j} = \max\left(|\Delta u_{i,j} + b_{i,j}| - \frac{\alpha}{\theta}, 0\right) sign\left(\Delta u_{i,j} + b_{i,j}\right). \tag{2.43}$$

Lastly, we give the overall implementation of the AL model (2.3), where the Bregman iterative parameter b is updated accordingly.

Algorithm 2.4.2: Split Bregman algorithm for AL model (2.3)

1: ***function*** $\mathrm{AL}_{Denoising}(f)$
2: Initialisation: Set $(w; b) = 0$, $(\alpha, \theta) > 0$
3: ***repeat***
4: Compute u according to (2.42)
5: Compute w according to (2.43)
6: Update Bregman iterative parameter $b \leftarrow b + \Delta u - w$
7: ***until*** some stopping criterion is satisfied
8: ***return*** u
9: ***end function***

2.4.3 Split Bregman for bounded Hessian (BH)

To solve the BH model (2.4) using the split Bregman method, we transform it into the following multivariable energy functional

$$E\left(u, \boldsymbol{w}; \boldsymbol{b}\right) = \frac{1}{2}\int_{\Omega}\left(u - f\right)^2 + \alpha\int_{\Omega}|\boldsymbol{w}| + \frac{\theta}{2}\int_{\Omega}\left|\boldsymbol{w} - \nabla^2 u - \boldsymbol{b}\right|^2, \tag{2.44}$$

where $\boldsymbol{w} = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix} \in (\mathbb{R}^{M\times N})^4$ is introduced to replace the Hessian of the function u. $|w| = \sqrt{\sum_{1\le n\le 4}\left(w_n\right)^2}$ represents the Frobenius norm of matrix $\boldsymbol{w}$. By introducing the Bregman iteration parameter $\boldsymbol{b} = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \in (\mathbb{R}^{M\times N})^4$ and applying the Bregman distance technique [21], the constraint $\boldsymbol{w} = \nabla^2 u$ can be effectively enforced.

Using the same manner to optimise (2.44), we first fix the variable $\boldsymbol{w}$ for u

$$u + \theta div^2\left(\nabla^2 u\right) = f + \theta div^2\left(\boldsymbol{w} - \boldsymbol{b}\right). \tag{2.45}$$

The discretisation form of (2.45) is given as follows according to the definition (2.24) and (2.30) in section 2.3

$$u_{i,j} + \theta\left(\partial_x^+\partial_x^-\partial_x^-\partial_x^+ u_{i,j} + \partial_y^-\partial_x^-\partial_y^+\partial_x^+ u_{i,j} + \partial_x^-\partial_y^-\partial_x^+\partial_y^+ u_{i,j} + \partial_y^+\partial_y^-\partial_y^-\partial_y^+ u_{i,j}\right) = G_{i,j}, \tag{2.46}$$

with $G = f + \theta\left(\partial_x^+\partial_x^-\left(w_1 - b_1\right) + \partial_y^-\partial_x^-\left(w_2 - b_2\right) + \partial_x^-\partial_y^-\left(w_3 - b_3\right) + \partial_y^+\partial_y^-\left(w_4 - b_4\right)\right)$. Applying the discrete Fourier transform to the both sides of the equation (2.46) gives

$$\mathcal{F}\left(u_{i,j} + \theta\left(\partial_x^+\partial_x^-\partial_x^-\partial_x^+ u_{i,j} + \partial_y^-\partial_x^-\partial_y^+\partial_x^+ u_{i,j} + \partial_x^-\partial_y^-\partial_x^+\partial_y^+ u_{i,j} + \partial_y^+\partial_y^-\partial_y^-\partial_y^+ u_{i,j}\right)\right) = \mathcal{F}\left(G_{i,j}\right). \tag{2.47}$$

The equivalent of (2.47) reads

$$\underbrace{\left(1 + 4\theta\left(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2\right)^2\right)}_{\chi}\mathcal{F}\left(u_{i,j}\right) = \mathcal{F}\left(G_{i,j}\right). \tag{2.48}$$

Note that the left-hand side of (2.48) is the same as that of (2.42). (2.48) gives the following closed-form

$$u_{i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(G_{i,j}\right)}{\chi}\right)\right). \tag{2.49}$$

Algorithm 2.4.3: Split Bregman algorithm for BH model (2.4)

1: ***function*** $\mathrm{BH}_{Denoising}(f)$
2: Initialisation: Set $(\boldsymbol{w}; \boldsymbol{b}) = 0$ and $(\alpha, \theta) > 0$
3: ***repeat***
4: Compute u according to (2.49)
5: Compute $\boldsymbol{w}$ according to (2.50)
6: Update Bregman iterative parameter $\boldsymbol{b} \leftarrow \boldsymbol{b} + \nabla^2 u - \boldsymbol{w}$
7: ***until*** some stopping criterion is satisfied
8: ***return*** u
9: ***end function***

Now fixing the variable u to derive the following Euler-Lagrange equation with respect to the 4D vector variable $\boldsymbol{w}$

$$\alpha\frac{\boldsymbol{w}}{|\boldsymbol{w}|} + \theta\left(\boldsymbol{w} - \nabla^2 u - \boldsymbol{b}\right) = 0,$$

which can be analytically solved by the discrete 4D generalised soft thresholding equation with the convention that $0/0 = 0$

$$\boldsymbol{w}_{i,j} = \max\left(\left|\nabla^2 u_{i,j} + \boldsymbol{b}_{i,j}\right| - \frac{\alpha}{\theta}, 0\right) \frac{\nabla^2 u_{i,j} + \boldsymbol{b}_{i,j}}{|\nabla^2 u_{i,j} + \boldsymbol{b}_{i,j}|}. \tag{2.50}$$

2.4.4 Split Bregman for hybridised TV and Laplacian (TVL)

As the TVL model (2.6) includes both the first and second order derivatives, we should introduce two splitting variables $\boldsymbol{w} \in (\mathbb{R}^{M\times N})^2$, $v \in \mathbb{R}^{M\times N}$, two Bregman iterative parameters $\boldsymbol{b} \in (\mathbb{R}^{M\times N})^2$, $d \in \mathbb{R}^{M\times N}$, and two penalty parameters (θ_1, θ_2), transforming the original functional (2.6) into

$$\begin{aligned} E\left(u, \boldsymbol{w}, v; \boldsymbol{b}, d\right) = \frac{1}{2}\int_\Omega \left(f-u\right)^2 \\ + \alpha \int_\Omega |\boldsymbol{w}| + \frac{\theta_1}{2}\int_\Omega |\boldsymbol{w} - \nabla u - \boldsymbol{b}|^2. \\ + \beta \int_\Omega |v| + \frac{\theta_2}{2}\int_\Omega \left(v - \Delta u - d\right)^2 \end{aligned}$$

The Euler-Lagrange equations with respect to u is

$$u - \theta_1 div\left(\nabla u\right) + \theta_2 \Delta\left(\Delta u\right) = f - \theta_1 div\left(\boldsymbol{w} - \boldsymbol{b}\right) + \theta_2 \Delta\left(v - d\right). \tag{2.51}$$

This formulation is a combination of (2.33) and (2.38), which can be also solved by the discrete Fourier transform. Its analytical solution is given as

$$u_{i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(G_{i,j}\right)}{\xi}\right)\right), \tag{2.52}$$

where G is the right-hand side of (2.51). $\xi = \xi_1 + \xi_2 - 1$ where ξ_1 and ξ_2 are (2.53) and (2.54), respectively.

$$\xi_1 = 1 - 2\theta_1\left(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2\right), \tag{2.53}$$

$$\xi_2 = 1 + 4\theta_2\left(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2\right)^2. \tag{2.54}$$

The discretisation problems of $\boldsymbol{w}$ and v are given by the following two equations respectively

$$\boldsymbol{w}_{i,j} = \max\left(|\nabla u_{i,j} + \boldsymbol{b}_{i,j}| - \frac{\alpha}{\theta_1}, 0\right) \frac{\nabla u_{i,j} + \boldsymbol{b}_{i,j}}{|\nabla u_{i,j} + \boldsymbol{b}_{i,j}|}, \tag{2.55}$$

$$v_{i,j} = \max\left(|\Delta u_{i,j} + d_{i,j}| - \frac{\beta}{\theta_2}, 0\right) sign\left(\Delta u_{i,j} + d_{i,j}\right). \tag{2.56}$$

Algorithm 2.4.4: Split Bregman algorithm for TVL model (2.6)

function $\text{TVL}_{Denoising}(f)$
Initialisation: Set $(\boldsymbol{w}, v; \boldsymbol{b}, d) = 0$, $(\alpha, \beta, \theta_1, \theta_2) > 0$
repeat
  Compute u according to (2.52)
  Compute $\boldsymbol{w}$ according to (2.55)
  Compute v according to (2.56)
  Update Bregman iterative parameter $\boldsymbol{b} \leftarrow \boldsymbol{b} + \nabla u - \boldsymbol{w}$
  Update Bregman iterative parameter $d \leftarrow d + \Delta u - v$
until some stopping criterion is satisfied
return u
end function

2.4.5 Split Bregman for hybridised TV and BH (TVBH)

Using the same manner as for the TVL model, we first transform the original (2.7) model into the following multivariable formulation

$$\begin{aligned} E\left(u, \boldsymbol{w}, \boldsymbol{v}; \boldsymbol{b}, \boldsymbol{d}\right) = \frac{1}{2}\int_\Omega (f-u)^2 \\ + \alpha\int_\Omega |\boldsymbol{w}| + \frac{\theta_1}{2}\int_\Omega |\boldsymbol{w} - \nabla u - \boldsymbol{b}|^2. \\ + \beta\int_\Omega |\boldsymbol{v}| + \frac{\theta_2}{2}\int_\Omega \left|\boldsymbol{v} - \nabla^2 u - \boldsymbol{d}\right|^2 \end{aligned}$$

To minimise the functional, we first fix $\boldsymbol{w} \in (\mathbb{R}^{M\times N})^2, \boldsymbol{v} \in (\mathbb{R}^{M\times N})^4, \boldsymbol{b} \in (\mathbb{R}^{M\times N})^2, \boldsymbol{d} \in (\mathbb{R}^{M\times N})^4$ for u and obtain its Euler-Lagrange equations

$$u - \theta_1 div\left(\nabla u\right) + \theta_2 div^2\left(\nabla^2 u\right) = f - \theta_1 div\left(\boldsymbol{w} - \boldsymbol{b}\right) + \theta_2 div^2\left(\boldsymbol{v} - \boldsymbol{d}\right). \tag{2.57}$$

This formulation is a combination of (2.33) and (2.45), which can be solved via the discrete fast Fourier transform. Its closed-form solution is given as

$$u_{i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(G_{i,j}\right)}{\xi}\right)\right), \tag{2.58}$$

where G is the right-hand side of (2.57). $\xi = \xi_1 + \xi_2 - 1$ where ξ_1 and ξ_2 are (2.53) and (2.54), respectively.

The solution of the variable $\boldsymbol{w}$ is same as (2.55), and the vector $\boldsymbol{v}$ can be updated by

$$\boldsymbol{v}_{i,j} = \max\left(\left|\nabla^2 u_{i,j} + \boldsymbol{d}_{i,j}\right| - \frac{\beta}{\theta_2}, 0\right) \frac{\nabla^2 u_{i,j} + \boldsymbol{d}_{i,j}}{|\nabla^2 u_{i,j} + \boldsymbol{d}_{i,j}|}. \tag{2.59}$$

Algorithm 2.4.5: Split Bregman algorithm for TVBH model (2.7)

01: ***function*** $\text{TVBH}_{Denoising}(f)$
02: Initialisation: Set $(\boldsymbol{w}, \boldsymbol{v}; \boldsymbol{b}, \boldsymbol{d}) = 0$, $(\alpha, \beta, \theta_1, \theta_2) > 0$
03: ***repeat***
04: Compute u according to (2.58)
05: Compute $\boldsymbol{w}$ according to (2.55)
06: Compute $\boldsymbol{v}$ according to (2.59)
07: Update Bregman iterative parameter $\boldsymbol{b} \leftarrow \boldsymbol{b} + \nabla u - \boldsymbol{w}$
08: Update Bregman iterative parameter $\boldsymbol{d} \leftarrow \boldsymbol{d} + \nabla^2 u - \boldsymbol{v}$
09: ***until*** some stopping criterion is satisfied
10: ***return*** u
11: ***end function***

2.4.6 Split Bregman for total generalised variation (TGV)

In order to solve the TGV model (2.8) using the split Bregman method, we introduce the auxiliary variables $(\boldsymbol{w}, \boldsymbol{v}; \boldsymbol{b}, \boldsymbol{d})$ and positive penalty parameters (θ_1, θ_2), transforming it to

$$\begin{aligned} E(u, \boldsymbol{p}, \boldsymbol{w}, \boldsymbol{v}; \boldsymbol{b}, \boldsymbol{d}) = \frac{1}{2}\int_\Omega (u - f)^2 \\ + \alpha \int_\Omega |\boldsymbol{w}| + \frac{\theta_1}{2}\int_\Omega |\boldsymbol{w} - \nabla u + \boldsymbol{p} - \boldsymbol{b}|^2. \\ + \beta \int_\Omega |\boldsymbol{v}| + \frac{\theta_2}{2}\int_\Omega |\boldsymbol{v} - \varepsilon(\boldsymbol{p}) - \boldsymbol{d}|^2 \end{aligned} \tag{2.60}$$

The meaning of the primal and dual variables in (2.60) is listed as follows:

- $u \in \mathbb{R}^{M\times N}$ denotes the denoised image we need to find;
- $\boldsymbol{p} = (p_1\ p_2) \in (\mathbb{R}^{M\times N})^2$ is the symmetrised gradient of the deformation field;
- $\boldsymbol{w} = (w_1\ w_2) \in (\mathbb{R}^{M\times N})^2$ is a 2D vector valued function related to the vector field $\nabla u - \boldsymbol{p}$;
- $\boldsymbol{v} = \begin{pmatrix} v_{11} & v_3 \\ v_3 & v_{22} \end{pmatrix} \in (\mathbb{R}^{M\times N})^4$ is a 4D matrix valued function related to the symmetrised derivative $\varepsilon(\boldsymbol{p})$;
- $\boldsymbol{b} = (b_1\ b_2) \in (\mathbb{R}^{M\times N})^2$ denotes the Bregman iterative parameter for enforcing $\boldsymbol{w} =$

$\nabla u - \boldsymbol{p}$;

• $\boldsymbol{d} = \begin{pmatrix} d_{11} & d_3 \\ d_3 & d_{22} \end{pmatrix} \in (\mathbb{R}^{M\times N})^4$ denotes the Bregman iterative parameter for enforcing $\boldsymbol{v} = \varepsilon(\boldsymbol{p})$.

Applying the alternating optimisation method, we first fix the variables $(\boldsymbol{p}, \boldsymbol{w}, \boldsymbol{v}; \boldsymbol{b}, \boldsymbol{d})$ to obtain the following Euler-Lagrange equations with respect to u

$$u - \theta_1 div(\nabla u) = f - \theta_1 div(\boldsymbol{w} + \boldsymbol{p} - \boldsymbol{b}). \tag{2.61}$$

The solution of this linear PDE is given as

$$u_{i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}(G_{i,j})}{\xi_1}\right)\right), \tag{2.62}$$

where G is the right-hand side of (2.61) and ξ_1 is defined as (2.53).

Fixing the variables $(u, \boldsymbol{w}, \boldsymbol{v}; \boldsymbol{b}, \boldsymbol{d})$ next, the Euler-Lagrange equations with respect to $\boldsymbol{p} = (p_1\ p_2)$ can be derived. For each p_1 and p_2, we have the following two discrete formulations

$$\left(\theta_1 - \theta_2\partial_x^+\partial_x^- - \frac{\theta_2}{2}\partial_y^+\partial_y^-\right)p_{1i,j} - \frac{\theta_2}{2}\partial_y^+\partial_x^- p_{2i,j} = h_{1i,j}, \tag{2.63}$$

$$\left(\theta_1 - \frac{\theta_2}{2}\partial_x^+\partial_x^- - \theta_2\partial_y^+\partial_y^-\right)p_{2i,j} - \frac{\theta_2}{2}\partial_x^+\partial_y^- p_{1i,j} = h_{2i,j}, \tag{2.64}$$

where the differential operators in (2.63) and (2.64) can be found in section 2.3 and

$$h_1 = \theta_1\left(\partial_x^+ u + b_1 - w_1\right) - \theta_2\partial_x^+(v_{11} - d_{11}) - \theta_2\partial_y^+(v_3 - d_3),$$

$$h_2 = \theta_1\left(\partial_y^+ u + b_2 - w_2\right) - \theta_2\partial_y^+(v_{22} - d_{22}) - \theta_2\partial_x^+(v_3 - d_3).$$

By applying the discrete Fourier transform to both sides of (2.63) and (2.64), we have the following system of linear equations

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} \mathcal{F}(p_{1i,j}) \\ \mathcal{F}(p_{2i,j}) \end{pmatrix} = \begin{pmatrix} \mathcal{F}(h_{1i,j}) \\ \mathcal{F}(h_{2i,j}) \end{pmatrix},$$

where the coefficients are

$$
\begin{aligned}
a_{11} &= \theta_1 - \theta_2\left(2\cos\frac{2\pi s}{N} - 2\right) - \frac{\theta_2}{2}\left(2\cos\frac{2\pi r}{M} - 2\right) \\
a_{12} &= -\frac{\theta_2}{2}\left(-1 + \cos\frac{2\pi r}{M} + \sqrt{-1}\sin\frac{2\pi r}{M}\right)\left(1 - \cos\frac{2\pi s}{N} + \sqrt{-1}\sin\frac{2\pi s}{N}\right) \\
a_{21} &= -\frac{\theta_2}{2}\left(-1 + \cos\frac{2\pi s}{N} + \sqrt{-1}\sin\frac{2\pi s}{N}\right)\left(1 - \cos\frac{2\pi r}{M} + \sqrt{-1}\sin\frac{2\pi r}{M}\right) \\
a_{22} &= \theta_1 - \frac{\theta_2}{2}\left(2\cos\frac{2\pi s}{N} - 2\right) - \theta_2\left(2\cos\frac{2\pi r}{M} - 2\right)
\end{aligned}
$$

Each component of the coefficient matrix $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ is a $M \times N$ sized matrix, and the determinant of the coefficient matrix is

$$
D = \left(\theta_1 - 2\theta_2\left(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2\right)\right)\left(\theta_1 - \theta_2\left(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2\right)\right),
$$

which is always positive for all discrete frequencies if $(\theta_1, \theta_2) > 0$. After the systems of linear equations are solved for each frequency r and s over the discrete frequency domain, we use the discrete inverse Fourier transform to obtain the analytical forms of p_1 and p_2

$$
p_{1i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{a_{22}\mathcal{F}\left(h_{1i,j}\right) - a_{12}\mathcal{F}\left(h_{2i,j}\right)}{D}\right)\right), \tag{2.65}
$$

$$
p_{2i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{a_{11}\mathcal{F}\left(h_{2i,j}\right) - a_{21}\mathcal{F}\left(h_{1i,j}\right)}{D}\right)\right). \tag{2.66}
$$

Fixing the variables $(u, \boldsymbol{p}, \boldsymbol{v}; \boldsymbol{b}, \boldsymbol{d})$ for the 2D vector $\boldsymbol{w}$, we have

$$
\alpha\frac{\boldsymbol{w}}{|\boldsymbol{w}|} + \theta_1\left(\boldsymbol{w} - \nabla u + \boldsymbol{p} - \boldsymbol{b}\right) = 0,
$$

which can be solved by the following discrete 2D analytical generalised soft thresholding equation with convention that $0/0 = 0$

$$
\boldsymbol{w}_{i,j} = \max\left(\left|\nabla u_{i,j} - \boldsymbol{p}_{i,j} + \boldsymbol{b}_{i,j}\right| - \frac{\alpha}{\theta_1}, 0\right)\frac{\nabla u_{i,j} - \boldsymbol{p}_{i,j} + \boldsymbol{b}_{i,j}}{\left|\nabla u_{i,j} - \boldsymbol{p}_{i,j} + \boldsymbol{b}_{i,j}\right|}. \tag{2.67}
$$

After $\boldsymbol{w}$ is solved, the Euler-Lagrange equations with respect to $\boldsymbol{v}$ is given as

$$
\beta\frac{\boldsymbol{v}}{|\boldsymbol{v}|} + \theta_2\left(\boldsymbol{v} - \varepsilon\left(\boldsymbol{p}\right) - \boldsymbol{d}\right) = 0,
$$

whose solution is following 4D analytical generalised soft thresholding equation with the convention that $0/0 = 0$

$$\boldsymbol{v}_{i,j} = \max\left(\left|\varepsilon\left(\boldsymbol{p}_{i,j}\right) + \boldsymbol{d}_{i,j}\right| - \frac{\beta}{\theta_2}, 0\right) \frac{\varepsilon\left(\boldsymbol{p}_{i,j}\right) + \boldsymbol{d}_{i,j}}{\left|\varepsilon\left(\boldsymbol{p}_{i,j}\right) + \boldsymbol{d}_{i,j}\right|}. \tag{2.68}$$

Finally, the overall **Algorithm** 2.4.6 is presented as

Algorithm 2.4.6: Split Bregman algorithm for TGV model (2.8)

function $\mathrm{TGV}_{Denoising}(f)$
Initialisation: Set $(\boldsymbol{p}, \boldsymbol{w}, \boldsymbol{v}; \boldsymbol{b}, \boldsymbol{d}) = 0$, $(\alpha, \beta, \theta_1, \theta_2) > 0$
repeat
  Compute u according to (2.62)
  Compute $\boldsymbol{p}$ according to (2.65) and (2.66)
  Compute $\boldsymbol{w}$ according to (2.67)
  Compute $\boldsymbol{v}$ according to (2.68)
  Update Bregman iterative parameter $\boldsymbol{b} \leftarrow \boldsymbol{b} + \nabla u - \boldsymbol{p} - \boldsymbol{w}$
  Update Bregman iterative parameter $\boldsymbol{d} \leftarrow \boldsymbol{d} + \varepsilon(\boldsymbol{p}) - \boldsymbol{v}$
until some stopping criterion is satisfied
return u
end function

2.4.7 Split Bregman for absolute curvature (AC)

In order to solve the AC model (2.10), we first consider the following splitting

$$E(u) = \frac{1}{2}\int_\Omega (u - f)^2 dx + \alpha \int_\Omega |q|\, dx. \tag{2.69}$$

$$\text{s.t. } q = div\left(\boldsymbol{n}\right), \boldsymbol{p} = \nabla u, \boldsymbol{n} = \boldsymbol{p}/|\boldsymbol{p}|$$

We note that constraint $\boldsymbol{n} = \boldsymbol{p}/|\boldsymbol{p}|$ is equivalent to the following two constraints according to the well-known Hölder inequality

$$|\boldsymbol{n}| \leq 1,\ |\boldsymbol{p}| = \boldsymbol{n} \cdot \boldsymbol{p}. \tag{2.70}$$

A new auxiliary vector $\boldsymbol{m}$ is then introduced to replace variable $\boldsymbol{n}$ in (2.70), which can be deemed as a relaxation of $\boldsymbol{n}$ and FFT can be thus applied. After that, we have the following five constraints

$$q = div\left(\boldsymbol{n}\right),\ \boldsymbol{p} = \nabla u,\ |\boldsymbol{m}| \leq 1,\ |\boldsymbol{p}| = \boldsymbol{m} \cdot \boldsymbol{p},\ \boldsymbol{m} = \boldsymbol{n}. \tag{2.71}$$

For $|m| \leq 1$ in image domain, we have Ω, $|\boldsymbol{p}| \geq \boldsymbol{m} \cdot \boldsymbol{p}$. By introducing the auxiliary variables and penalty parameters, the split Bregman method can be employed to transform (2.69) with the constraints in (2.71) into the following constrained energy functional

$$\begin{aligned} E\left(u, q, \boldsymbol{p}, \boldsymbol{n}, \boldsymbol{m}; b_1, \boldsymbol{b_2}, b_3, \boldsymbol{b_4}\right) = & \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega |q| \\ & + \int_\Omega (\theta_1 + b_1)(|\boldsymbol{p}| - \boldsymbol{p}\cdot\boldsymbol{m}) + \frac{\theta_2}{2}\int_\Omega |\boldsymbol{p} - \nabla u - \boldsymbol{b_2}|^2, \\ & + \frac{\theta_3}{2}\int_\Omega (q - div(\boldsymbol{n}) - b_3)^2 + \frac{\theta_4}{2}\int_\Omega |\boldsymbol{n} - \boldsymbol{m} - \boldsymbol{b_4}|^2 \end{aligned} \tag{2.72}$$

$$\text{s.t.} \quad |\boldsymbol{m}| \leq 1.$$

Note that for the constraint $|\boldsymbol{p}| = \boldsymbol{m} \cdot \boldsymbol{p}$, we use the augmented Lagrangian method with L1-norm for the penalisation in the functional (2.72) as it is true that $|\boldsymbol{p}| - \boldsymbol{m} \cdot \boldsymbol{p} > 0$ when $|\boldsymbol{m}| \leq 1$. The meaning of the primal and dual variables in (2.72) is listed as in the following:

- $u \in \mathbb{R}^{M\times N}$ denotes the denoised image we need to find;
- $q \in \mathbb{R}^{M\times N}$ is a scalar valued function related to the divergence of vector function $\boldsymbol{n}$;
- $\boldsymbol{p} = (p_1\ p_2) \in (\mathbb{R}^{M\times N})^2$ is a vector valued function related to the gradient of the function u;
- $\boldsymbol{n} = (n_1\ n_2) \in (\mathbb{R}^{M\times N})^2$ is a vector valued function related to the unit vectors of the level curves of u;
- $\boldsymbol{m} = (m_1\ m_2) \in (\mathbb{R}^{M\times N})^2$ is a vector valued function used to relax $\boldsymbol{n}$ and thus FFT can be used;
- b_1 denotes a scalar Lagrangian multiplier for the constraint $|\boldsymbol{p}| = \boldsymbol{m} \cdot \boldsymbol{p}$;
- $\boldsymbol{b_2} = (b_{21}\ b_{22}) \in (\mathbb{R}^{M\times N})^2$ denotes a vector Bregman iterative parameter for the constraint $\boldsymbol{p} = \nabla u$;
- b_3 denotes a scalar Bregman iterative parameter for the constraint $q = div(\boldsymbol{n})$;
- $\boldsymbol{b_4} = (b_{41}\ b_{42}) \in (\mathbb{R}^{M\times N})^2$ denotes a vector Bregman iterative parameter for the constraint $\boldsymbol{m} = \boldsymbol{n}$.

Using the alternating optimisation technique and first fixing $(q, \boldsymbol{p}, \boldsymbol{n}, \boldsymbol{m}; b_1, \boldsymbol{b_2}, b_3, \boldsymbol{b_4})$, we obtain the Euler-Lagrange equations with respect to u as follows

$$u - \theta_2 div\left(\nabla u\right) = f - \theta_2 div\left(\boldsymbol{p} - \boldsymbol{b_2}\right). \tag{2.73}$$

The solution of this second order linear PDE is given as

$$u_{i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(G_{i,j}\right)}{\xi_1}\right)\right), \tag{2.74}$$

where G is the right-hand side of (2.73) and ξ_1 is defined as (2.53).

$(u, \boldsymbol{p}, \boldsymbol{n}, \boldsymbol{m}; b_1, \boldsymbol{b_2}, b_3, \boldsymbol{b_4})$ are fixed next, and we obtain the following equation

$$\alpha\frac{q}{|q|} + \theta_3\left(q - div\left(\boldsymbol{n}\right) - b_3\right) = 0.$$

Its solution reads as

$$q_{i,j} = \max\left(\left|div(\boldsymbol{n})_{i,j} + b_{3i,j}\right| - \frac{\alpha}{\theta_3}, 0\right) sign\left(div(\boldsymbol{n}_{i,j}) + b_{3i,j}\right). \tag{2.75}$$

Fixing $(u, q, \boldsymbol{n}, \boldsymbol{m}; b_1, \boldsymbol{b_2}, b_3, \boldsymbol{b_4})$, we obtain the Euler-Lagrange equations with respect to $\boldsymbol{p}$

$$(\theta_1 + b_1)\frac{\boldsymbol{p}}{|\boldsymbol{p}|} + \theta_2\left(\boldsymbol{p} - \nabla u - \boldsymbol{b_2}\right) - (\theta_1 + b_1)\boldsymbol{m} = 0,$$

which can be solved by the analytical soft thresholding equation as

$$\boldsymbol{p}_{i,j} = \max\left(|\boldsymbol{S}_{i,j}| - \frac{(\theta_1 + b_{1i,j})}{\theta_2}, 0\right)\frac{\boldsymbol{S}_{i,j}}{|\boldsymbol{S}_{i,j}|}, \tag{2.76}$$

where $\boldsymbol{S}_{i,j} = \nabla u_{i,j} + \boldsymbol{b_2}_{i,j} + \frac{(\theta_1 + b_{1i,j})\boldsymbol{m}_{i,j}}{\theta_2}$. If $|\boldsymbol{S}_{i,j}|$ is 0, we have $0 \cdot (0/0) = 0$.

Fixing $(u, q, \boldsymbol{p}, \boldsymbol{m}; b_1, \boldsymbol{b_2}, b_3, \boldsymbol{b_4})$, the following two Euler-Lagrange equationss for the vector variable $\boldsymbol{n} = (n_1\ n_2)$ are derived, respectively

$$\left(\theta_4 - \theta_3\partial_x^+\partial_x^-\right)n_1 - \theta_3\partial_x^+\partial_y^- n_2 = h_{1i,j}, \tag{2.77}$$

$$\left(\theta_4 - \theta_3\partial_y^+\partial_y^-\right)n_2 - \theta_3\partial_y^+\partial_x^- n_1 = h_{2i,j}, \tag{2.78}$$

where

$$h_1 = \theta_4\left(m_1 + b_{41}\right) - \theta_3\partial_x^+\left(q - b_3\right),$$

$$h_2 = \theta_4\left(m_2 + b_{42}\right) - \theta_3\partial_y^+\left(q - b_3\right).$$

By applying the discrete Fourier transform to both sides of (2.77) and (2.78), we have the following linear system

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} \mathcal{F}(n_{1i,j}) \\ \mathcal{F}(n_{2i,j}) \end{pmatrix} = \begin{pmatrix} \mathcal{F}(h_{1i,j}) \\ \mathcal{F}(h_{2i,j}) \end{pmatrix},$$

where the coefficients are

$$
\begin{aligned}
a_{11} &= \theta_4 - 2\theta_3\left(\cos\frac{2\pi s}{N} - 1\right) \\
a_{12} &= -\theta_3\left(1 - \cos\frac{2\pi r}{M} + \sqrt{-1}\sin\frac{2\pi r}{M}\right)\left(-1 + \cos\frac{2\pi s}{N} + \sqrt{-1}\sin\frac{2\pi s}{N}\right) \\
a_{21} &= -\theta_3\left(1 - \cos\frac{2\pi s}{N} + \sqrt{-1}\sin\frac{2\pi s}{N}\right)\left(-1 + \cos\frac{2\pi r}{M} + \sqrt{-1}\sin\frac{2\pi r}{M}\right) \\
a_{22} &= \theta_4 - 2\theta_3\left(\cos\frac{2\pi r}{M} - 1\right)
\end{aligned}.
$$

The coefficient matrix $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ is a $M \times N$ numbers of 2×2 system, whose determinant is

$$
D = \theta_4^2 - 2\theta_3\theta_4\left(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2\right)
$$

which is always positive for all discrete frequencies if $(\theta_3, \theta_4) > 0$. After the systems of linear equations are solved for each frequency r and s over the discrete frequency domain, we use the discrete inverse Fourier transform to obtain the analytical forms of n_1 and n_2

$$
n_{1i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{a_{22}\mathcal{F}\left(h_{1i,j}\right) - a_{12}\mathcal{F}\left(h_{2i,j}\right)}{D}\right)\right), \tag{2.79}
$$

$$
n_{2i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{a_{11}\mathcal{F}\left(h_{2i,j}\right) - a_{21}\mathcal{F}\left(h_{1i,j}\right)}{D}\right)\right). \tag{2.80}
$$

Fixing $(u, q, \boldsymbol{p}, \boldsymbol{n}; b_1, \boldsymbol{b_2}, b_3, \boldsymbol{b_4})$, $\boldsymbol{m}$ can be derived from the following closed-form

$$
\tilde{\boldsymbol{m}}_{i,j} = \frac{\left(\theta_1 + b_1\right)\boldsymbol{p}_{i,j}}{\theta_4} + \boldsymbol{n}_{i,j} - \boldsymbol{b_4}_{i,j}.
$$

In order to satisfy the constraint $|\boldsymbol{m}| \leq 1$, the following projection formulation is then imposed on $\tilde{\boldsymbol{m}}_{i,j}$

$$
\boldsymbol{m}_{i,j} = \frac{\tilde{\boldsymbol{m}}_{i,j}}{\max(|\tilde{\boldsymbol{m}}_{i,j}|, 1)}. \tag{2.81}
$$

Finally, we update the auxiliary variables $(b_1, \boldsymbol{b_2}, b_3, \boldsymbol{b_4})$, which are listed from the step 09 to 12 in **Algorithm** 2.4.7. The split Bregman algorithm for the AC denoising problem (2.10) is as follows

Algorithm 2.4.7: Split Bregman algorithm for AC model (2.10)

function $\mathrm{AC}_{Denoising}(f)$
Initialisation: Set $(q, \boldsymbol{p}, \boldsymbol{n}, \boldsymbol{m}; b_1, \boldsymbol{b_2}, b_3, \boldsymbol{b_4}) = 0$, $(\alpha, \theta_1, \theta_2, \theta_3, \theta_4) > 0$
repeat
  Compute u according to (2.74)
  Compute q according to (2.75)
  Compute $\boldsymbol{p}$ according to (2.76)
  Compute $\boldsymbol{n}$ according to (2.79) and (2.80)
  Compute $\boldsymbol{m}$ according to (2.81)
  Update Lagrangian multiplier $b_1 \leftarrow b_1 + \theta_1(|\boldsymbol{p}| - \boldsymbol{m} \cdot \boldsymbol{p})$
  Update Bregman iterative parameter $\boldsymbol{b_2} \leftarrow \boldsymbol{b_2} + \nabla u - \boldsymbol{p}$
  Update Bregman iterative parameter $b_3 \leftarrow b_3 + div(\boldsymbol{n}) - q$
  Update Bregman iterative parameter $\boldsymbol{b_4} \leftarrow \boldsymbol{b_4} + \boldsymbol{m} - \boldsymbol{n}$
until some stopping criterion is satisfied
return u
end function

2.5 Numerical experiments

In this section, the performance of different image denoising models is compared quantitatively and qualitatively. The models and their energy functionals are summarised in Table 2.1 for easy comparison. The metrics used for quantitative comparison are peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), root mean square error (RMSE), and structure similarity index map (SSIM). Experiments are performed using Matlab 2014b on Windows 7 with Intel Xeon CPU E51620 at 3.7GHz and 32GB memory.

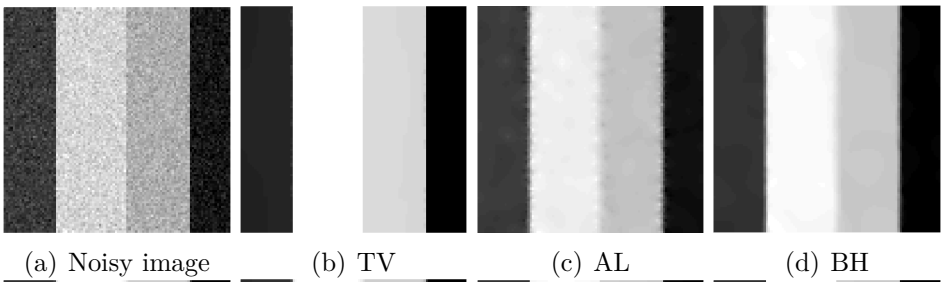
Table 2.1: Models for comparison

No.	Model	Energy functional
1	TV	$E(u) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega \lvert\nabla u\rvert$
2	AL	$E(u) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega \lvert\Delta u\rvert$
3	BH	$E(u) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega \lvert\nabla^2 u\rvert$
4	TVL	$E(u) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega \lvert\nabla u\rvert + \beta \int_\Omega \lvert\Delta u\rvert$
5	TVBH	$E(u) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega \lvert\nabla u\rvert + \beta \int_\Omega \lvert\nabla^2 u\rvert$
6	TGV	$E(u,p) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega \lvert\nabla u - p\rvert + \beta \int_\Omega \lvert\varepsilon(p)\rvert$
7	AC	$E(u) = \frac{1}{2}\int_\Omega (u-f)^2 + \alpha \int_\Omega \lvert div(\frac{\nabla u}{\lvert\nabla u\rvert})\rvert$

2.5.1 Comparison of edge and contrast preserving ability

In Figure 2.4, we test the edge preservation ability of different models on a piecewise constant image (a) through visual inspection. From (b) to (h), it is clear that the TV

and AC models perfectly maintain the edges of the strips. TGV also shows very good edge preservation result. However, it results in additional lines on the denoised stripes because it approximates the image with the piecewise affine function. (c) illustrates that directly applying the BH regulariser blurs the edges of objects, whilst (b) shows that the AL model using the Laplace regulariser is the worst approach for edge preservation. By incorporating the TV regulariser to the BH and AL models, TVL and TVBH improve the denoised quality, as shown in (e) and (f). However, the image boundaries have been slightly smeared by these two models.



(a) Noisy image (b) TV (c) AL (d) BH

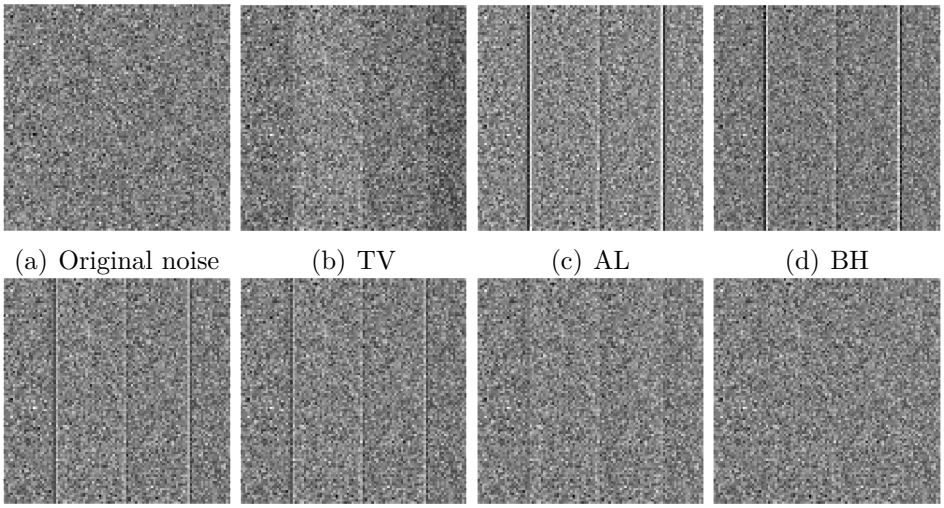
(e) TVL (f) TVBH (g) TGV (h) AC

Figure 2.4: Comparison of edge preserving ability. The methods in Table 2.1 are performed on the noisy image (a) and their corresponding denoised results are shown in (b)-(h), respectively.

In Figure 2.5, we present the associated residual images $f - u$ for all the models compared, where f is the input noisy data, as shown in Figure 2.4 (a), and u denotes each corresponding denoised image, as shown in Figure 2.4 (b) to (h). As can be seen from Figure 2.5, the residual (h) from the AC model contains almost all noise, whilst the rest of residual images contain more or less structural information of the noisy data. This indicates some contrast information has been lost in Figure 2.4 (b) to (g). The AC model is the best in terms of image contrast preservation.

In Figure 2.6, we show the middle slices of the denoised images in Figure 2.4. It can be seen from (h) that there is almost no difference between the denoised slice curve (red) and the noise free slice curve (blue). This demonstrates the capability of the AC model for image contrast and edge preservation. Due to the contrast loss, the red curve in (b) obtained by the TV model sightly deviates from the blue one in the vertical direction.

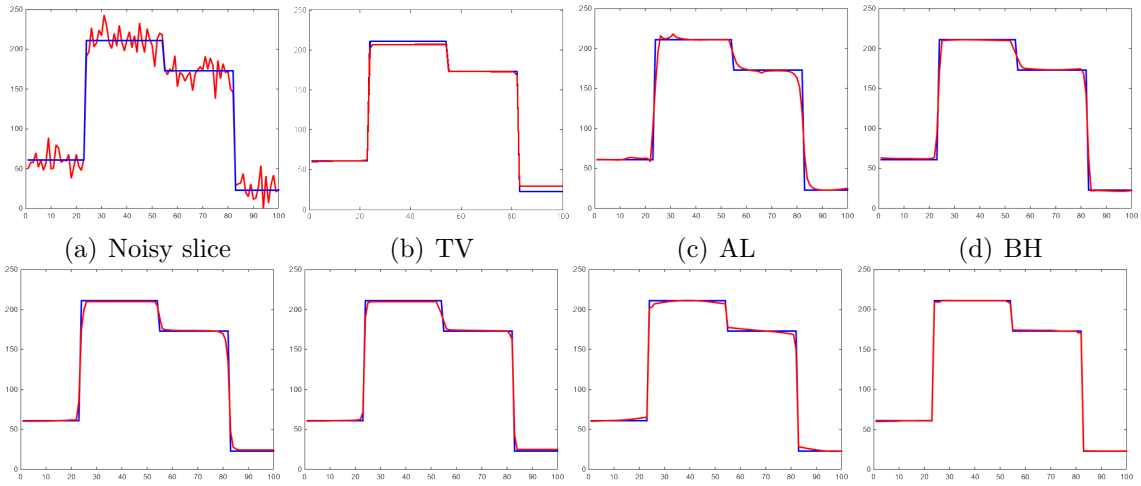
As compared to the TV and AC models, the rest of denoising models impose too much smoothness on their curves, leading to less pleasant fitting results.



(a) Original noise (b) TV (c) AL (d) BH

(e) TVL (f) TVBH (g) TGV (h) AC

Figure 2.5: Residual images of the denoised examples in Figure 2.4



(a) Noisy slice (b) TV (c) AL (d) BH

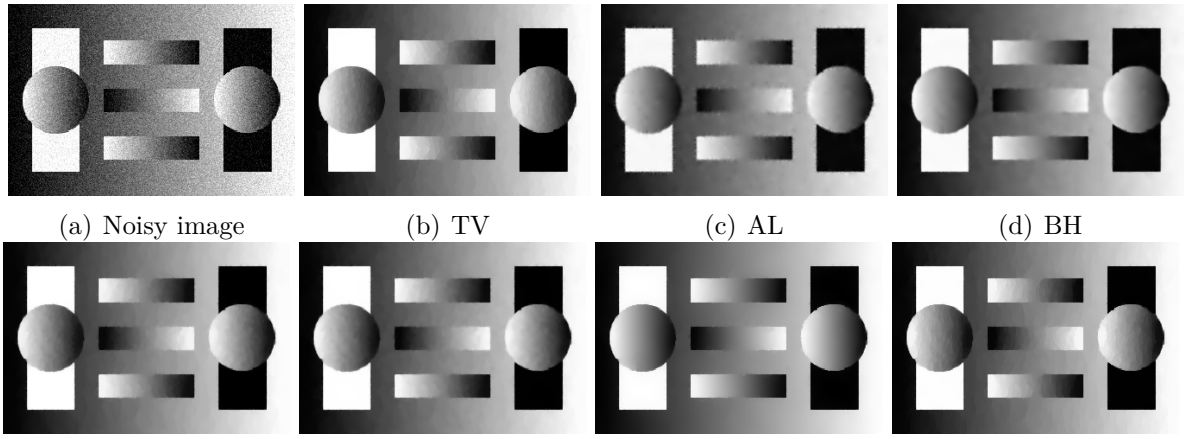
(e) TVL (f) TVBH (g) TGV (h) AC

Figure 2.6: Plotting the middle slices of the denoised images in Figure 2.4. The blue curve is the ground truth.

2.5.2 Comparison of smoothness preserving ability

We now test the smoothness preservation ability of different models on a piecewise smooth image. In order to do this, the clean image Figure 2.2 (a) is corrupted with the Gaussian

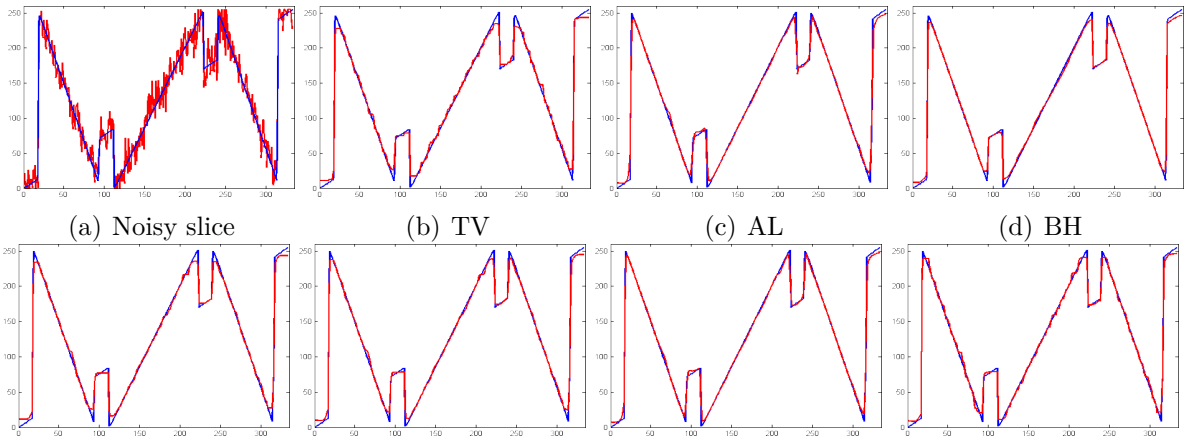
noise of 0.005 variance, and the corrupted image is shown in Figure 2.7 (a). The noise is produced with MATLAB's built-in function *imnoise*.



(a) Noisy image (b) TV (c) AL (d) BH

(e) TVL (f) TVBH (g) TGV (h) AC

Figure 2.7: Performance comparison of different methods on the noisy image (a).



(a) Noisy slice (b) TV (c) AL (d) BH

(e) TVL (f) TVBH (g) TGV (h) AC

Figure 2.8: Plotting the middle slices of the denoised images in Figure 2.7. Blue curve is the ground truth.

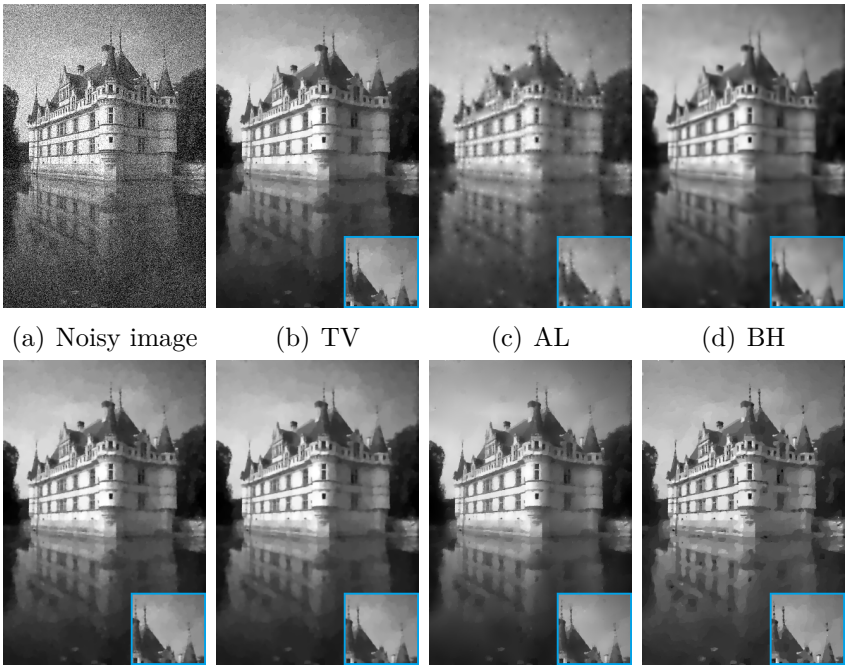
In Figure 2.7, the denoised results from (a) obtained by different methods are shown. As evident, the TV model leads to the staircase effect, which is confirmed in (b) where jagged appearance is seen. In addition, the cleaned results, as shown in (e) and (f), illustrate that the hybridised higher order models involving the TV regulariser also have such artefact. Further, the AC model is unable to remove the side effect entirely, as observed in (h). Although the AL and BH models appear to eliminate the unfavourable artefact, their denoised results are not very visually satisfactory. For the AL model, there exists some undesirable spikes at the image boundaries, whilst the BH model slightly

blurs the object edges. The visual effect in (g) shows that TGV completely gets rid of the artefact and meanwhile preserves the shape edges in the image. The values of PSNR, SNR, SSIM and RMSE shown in Table 2.2 quantitatively validate that TGV is the best for denoising the piecewise smooth image.

Table 2.2: Comparison of PSNR, SNR, RMSE and SSIM using different methods on the noisy image Figure 2.7 (a).

	Figure 2.7: noise variance 0.005.			
	PSNR	SNR	RMSE	SSIM
Noisy image	23.6978	19.1788	7.9693	0.3278
TV	32.6621	26.1032	3.6633	0.9015
AL	27.8122	23.2933	3.9571	0.8812
BH	30.0447	25.5257	3.7131	0.8985
TVL	30.6670	26.1480	3.4764	0.9049
TVBH	32.3548	27.8358	3.2725	0.9118
TGV	**34.4020**	**29.8830**	**3.2120**	**0.9243**
AC	34.2254	29.7064	3.2412	0.9039

In Figure 2.8, we plot the middle slices of the examples in Figure 2.7. Note that the results are overlapped with the ground truth curve, shown in blue. From the figure, one can see that the AL, BH and TGV models lead to smoother curve fitting results than the TV, TVL, TVBH and AC models. Among all the methods compared, TGV gives the best fitting result.



(a) Noisy image (b) TV (c) AL (d) BH

(e) TVL (f) TVBH (g) TGV (h) AC

Figure 2.9: Comparison of different methods on a real noisy image (a) (size 480×320).

2.5.3 Test on a real image

In this experiment, a real image is used to further evaluate different models. In order to make the denoising evaluation more objective and convincing, the clean image is corrupted by additive Gaussian noise with zero-mean and different variance. Six variance values from 0.005 to 0.03 at 0.005 intervals are used.

Table 2.3: Comparison of PSNR, SNR, RMSE and SSIM using different methods for different noise variance.

	PSNR test						SNR test					
Noise variance	0.005	0.01	0.015	0.02	0.025	0.03	0.005	0.01	0.015	0.02	0.025	0.03
Noisy image	23.1360	20.2415	18.5938	17.4469	16.5558	15.8704	16.4828	13.5883	11.9405	10.7936	9.9026	9.2171
TV	27.5911	26.7111	24.8781	24.5730	24.4611	24.3582	20.1908	19.2579	18.0248	17.8397	17.7578	17.5949
AL	26.8441	24.5273	23.4462	23.2245	23.1309	22.8787	20.1908	17.8740	16.7929	16.5712	16.4776	16.2254
BH	27.1086	25.5418	23.8114	23.6834	23.5913	23.4151	20.4554	18.8885	17.1582	17.0302	16.9380	16.7619
TVL	27.6490	26.8506	25.3305	25.1088	24.6049	24.3680	20.7958	19.8373	18.6773	18.4555	17.9517	17.7147
TVBH	28.8336	27.8218	25.9831	25.6466	25.0526	24.5616	22.1803	20.1685	19.3298	18.8680	18.3994	17.8083
TGV	**28.8697**	**28.4359**	**27.5416**	**27.0896**	**26.7111**	**26.5212**	**22.2064**	**20.8092**	**20.1619**	**19.9746**	**19.4216**	**19.0020**
AC	28.8479	28.0366	26.6016	25.9456	25.6491	24.9981	22.1824	20.1791	19.4009	18.9976	18.6411	18.0981
	RMSE test						SSIM test					
Noise variance	0.005	0.01	0.015	0.02	0.025	0.03	0.005	0.01	0.015	0.02	0.025	0.03
Noisy image	8.4469	9.2784	9.6300	9.8620	10.0313	10.1224	0.4166	0.3138	0.2617	0.2296	0.2067	0.1890
TV	5.1521	5.5854	5.9036	5.9226	5.9854	6.0803	0.8210	0.7901	0.7611	0.7416	0.7303	0.7259
AL	5.4106	5.7604	5.9893	6.0573	6.0969	6.1873	0.7962	0.7415	0.7059	0.6964	0.6896	0.6795
BH	5.2352	5.7011	5.9699	5.9932	6.0424	6.1005	0.8155	0.7739	0.7327	0.7179	0.7130	0.7055
TVL	5.0821	5.5556	5.8444	5.9194	5.9214	6.0699	0.8259	0.7926	0.7674	0.7551	0.7386	0.7302
TVBH	5.0036	5.4678	5.7115	5.8550	5.8968	6.0401	0.8412	0.8008	0.7781	0.7581	0.7469	0.7323
TGV	**4.9287**	**5.3898**	**5.5003**	**5.6452**	**5.7459**	**6.0159**	**0.8487**	**0.8109**	**0.7952**	**0.7794**	**0.7717**	**0.7600**
AC	4.9403	5.4081	5.5569	5.7095	5.8216	6.0299	0.8420	0.8015	0.7806	0.7601	0.7503	0.7465

Table 2.4: Comparison of time per iteration, total CPU time and total iterations of different models for 6 noise levels.

Noise variance		TV	AL	BH	TVL	TVBH	TGV	AC
0.005	Time per iteration (s)	0.0471	0.0390	0.0551	0.0581	0.0741	0.1003	0.1278
	Total CPU time (s)	2.6864	3.6391	3.4713	5.1863	3.6246	13.5646	7.6698
	Total iterations	62	100	63	96	57	147	70
0.01	Time per iteration (s)	0.0463	0.0340	0.0628	0.0515	0.0747	0.0956	0.1070
	Total CPU time (s)	2.7551	4.8098	3.4373	4.8496	3.9162	20.2070	22.8778
	Total iterations	61	121	57	90	61	218	215
0.015	Time per iteration (s)	0.0457	0.0386	0.0499	0.0566	0.0735	0.1034	0.1175
	Total CPU time (s)	2.6152	5.5070	2.8969	4.7385	4.3204	18.7648	12.6513
	Total iterations	61	156	55	91	67	204	117
0.02	Time per iteration (s)	0.0447	0.0368	0.0536	0.0576	0.0734	0.1113	0.1241
	Total CPU time (s)	2.9139	6.4935	3.4635	4.5916	5.6789	19.5086	8.3592
	Total iterations	62	177	65	87	86	212	77
0.025	Time per iteration (s)	0.0463	0.0379	0.0528	0.0577	0.0760	0.1083	0.1303
	Total CPU time (s)	2.5942	7.3905	3.4833	4.8650	5.5162	30.8093	14.2515
	Total iterations	60	203	69	90	83	335	130
0.03	Time per iteration (s)	0.0456	0.0368	0.0520	0.0552	0.0793	0.1102	0.1205
	Total CPU time (s)	2.6202	6.6830	3.9454	4.9202	5.9294	24.2398	10.4639
	Total iterations	62	185	74	91	92	262	96

In Figure 2.9, we show the denoising results of the noisy image (a) which contains the Gaussian noise with 0.015 variance. For each denoised image, the corresponding close-up region is displayed on the bottom right corner for detailed comparison. The

visual inspection illustrates that the TV, TVL, TVBH and AC models yield the staircase effect, whilst the AL and BH models smear the image edges. Again, TGV achieves a more pleasant result, entirely eliminating the artefact while nicely preserving the sharp boundaries. Hence, TGV used as a penalty functional prefers images which appear more natural and leads to a more faithful denoising for real images.

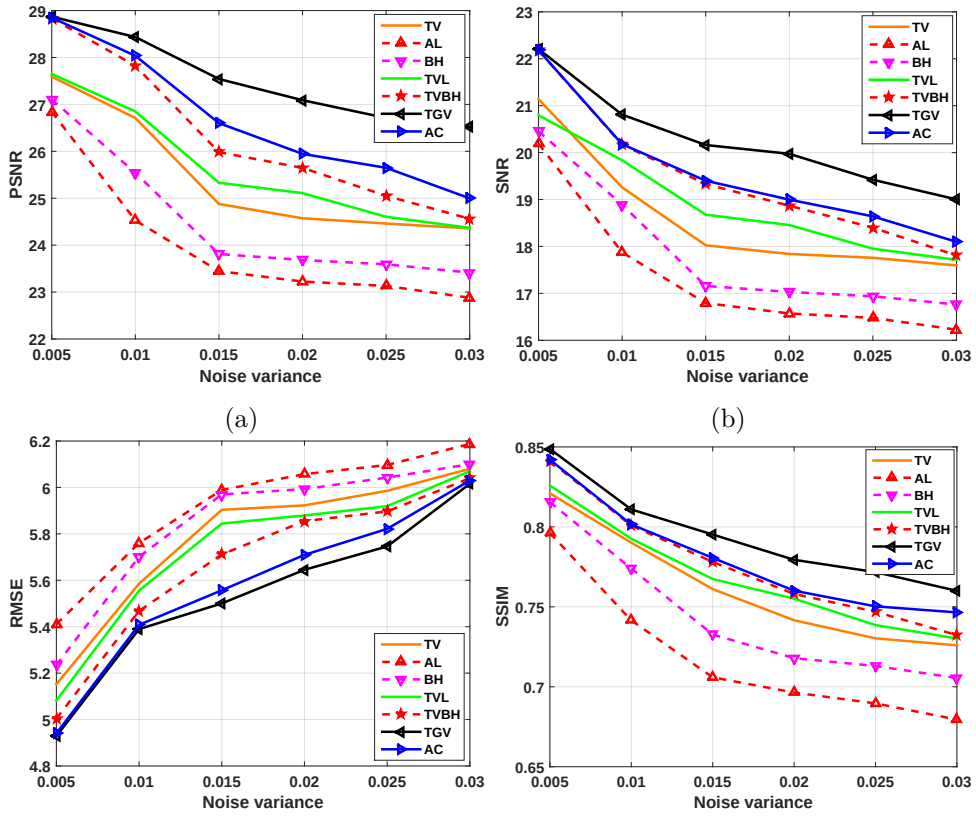


Figure 2.10: Plots of quantitative evaluation matrices for different methods on real data with different noise variance. (a) PSNR index; (b) SNR index; (c) RMSE index; (d) SSIM index.

In Table 2.3, the performance metrics are shown to quantitatively compare different methods for 6 noise levels. In Figure 2.10 we plot the values of PSNR, SNR, RMSE and SSIM against increasing levels of noise. From these, we can rank the models for denoising a real image in decreasing order as TGV>AC>TVBH>TVL>TV>BH>AL.

Table 2.4 demonstrates the computational efficiency of the different models. We set stopping criteria for all as $|E^k - E^{k-1}|/E^k \leq \epsilon$, where E is the value of energy functional of each model, k is the current iteration and ϵ is a small tolerance used to stop the iteration. $\epsilon = 10^{-5}$ in all cases. Statistical analysis of Table 2.4 is shown as boxplots in Figure 2.11, where each box represents variations of the denoising results of a model for

6 noise levels. It is clear that TGV is the slowest, TV is the fastest, and AL, TGV and AC are less efficient than BH, TVL and TVBH.

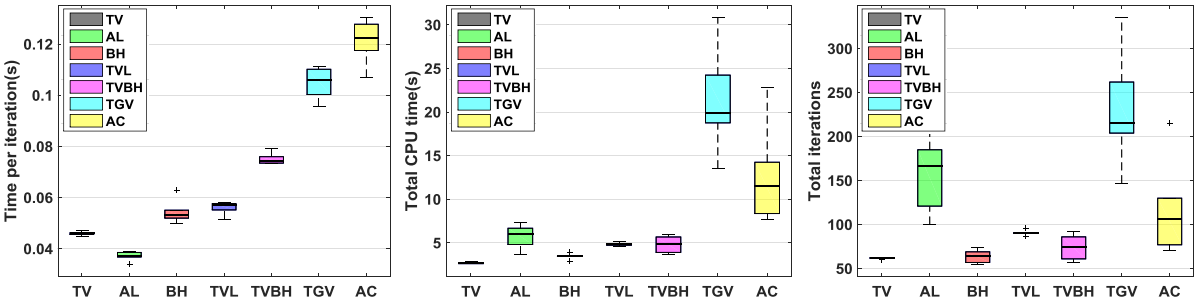


Figure 2.11: Boxplots of the experimental data in Table 2.4. (a) Boxplot of time per iteration for 6 noise levels; (b) Boxplot of total CPU time for 6 noise levels; (c) Boxplot of total iterations for 6 noise levels.

Table 2.5: Computational speed comparison over different methods with a fixed time budget, i.e., $\mathrm{T} = 1s, 2s, 3s, 4s, 5s, 10s$. In each time budget, the maximum iteration number (mean±standard deviation in 5 runs) of different methods is recorded.

Method	Iter. # ($1s$)	Iter. # ($2s$)	Iter. # ($3s$)	Iter. # ($4s$)	Iter. # ($5s$)	Iter. # ($10s$)
TV	39.40 ± 2.2	83.40 ± 6.3	128.6 ± 4.0	170.2 ± 9.7	214.8 ± 18.6	444.2 ± 12.1
AL	**58.80 ± 3.9**	**103.4 ± 6.2**	**165.8 ± 4.4**	**214.8 ± 8.6**	**272.2 ± 14.2**	**531.0 ± 20.7**
BH	35.00 ± 1.6	69.20 ± 2.5	107.2 ± 1.9	142.0 ± 2.5	181.8 ± 2.90	348.4 ± 22.6
TVL	15.00 ± 0.7	29.80 ± 1.3	46.00 ± 2.5	58.80 ± 1.3	165.0 ± 4.20	325.2 ± 14.0
TVBH	25.60 ± 1.1	51.40 ± 0.5	76.80 ± 1.3	101.4 ± 4.9	127.8 ± 2.80	156.2 ± 3.40
TGV	9.600 ± 0.5	19.20 ± 0.4	27.60 ± 2.1	37.80 ± 0.8	47.20 ± 0.80	94.00 ± 3.40
AC	8.800 ± 1.1	20.00 ± 0.7	29.20 ± 1.3	39.40 ± 1.5	49.80 ± 1.90	97.80 ± 2.30

In this experiment, we further compare the speed of different methods with a fixed time budget. In Table 2.5 and its corresponding plot (Figure 2.12 left), we show how many iterations each algorithm can reach within time $\mathrm{T} = 1s, 2s, 3s, 4s, 5s, 10s$. It is obvious that the split Bregman algorithm for the TV and TL models runs the most iterations, whilst the split Bregman for the TGV and AC models are the two slowest methods among all compared. In addition, the hybridised models (i.e. TVL and TVBH) are less efficient than their counterparts (i.e. AL and BH). The computational speed

Table 2.6: Comparison of root mean square error (RMSE) produced by different methods in each time budget, i.e., $\mathrm{T} = 1s, 2s, 3s, 4s, 5s, 10s$.

	RMSE ($1s$)	RMSE ($2s$)	RMSE ($3s$)	RMSE ($4s$)	RMSE ($5s$)	RMSE ($10s$)
TV	40.1639	40.1772	40.1784	40.1787	40.1786	40.1788
TL	40.3041	40.3277	40.3373	40.3427	40.3455	40.3490
BH	33.7715	33.7590	33.7592	33.7604	33.7611	33.7655
TVL	39.3741	39.4300	39.4381	39.4406	39.4433	39.4467
TVBH	14.2322	14.0752	13.7411	13.5165	13.4761	13.4410
TGV	**13.1190**	**12.6023**	**12.5028**	**11.7634**	**11.7782**	**11.0737**
TC	14.2470	12.9305	12.5531	12.4920	12.4850	12.4711

comparison in this experiment shows a same consistency with that in Table 2.4 and Figure 2.11

Lastly, in Table 2.6 and its corresponding plot (Figure 2.12 right), we compare the root mean square error (RMSE) produced by each method in different time budgets, i.e., T $= 1s, 2s, 3s, 4s, 5s, 10s$. As can be seen, as iteration proceeds their RMSEs drops gradually. However, due to the fast FFT-based split Bregman implementation, these models have achieved stable results after $1s$. That is why the plots shown in Figure 2.12 right have little changes at different time points. In addition, TVG has a slower convergence rate as compared to other methods, as its RMSE is still decreasing. Note that the quantitative numbers are acquired by testing different methods on the noisy image Figure 2.9 (a) with 0.015 variance Gaussian noise

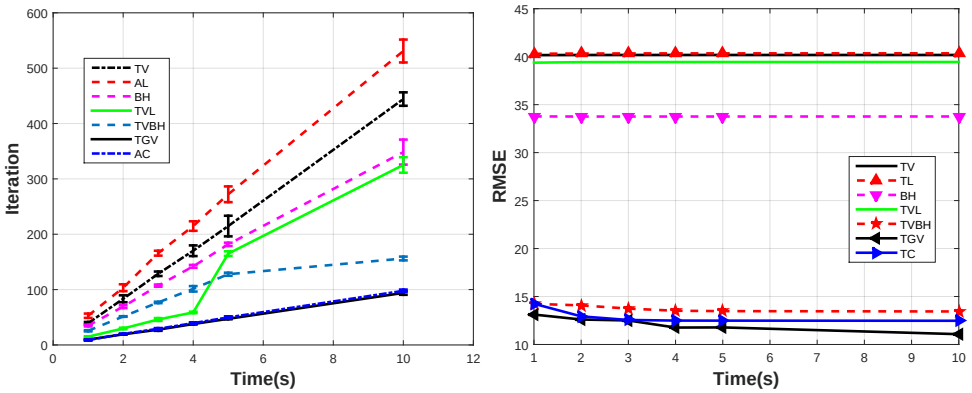


Figure 2.12: Plots of the experimental data in Table 2.5 and 2.6. Left: maximum iteration number (mean±standard deviation in 5 runs) versus different time budgets. Right: RMSE versus different time budgets.

2.6 Summary

In this chapter, several first and second order variational models have been studied and their application for image denoising has been demonstrated. Detailed descretisation process based on the finite difference scheme is given, so is the numerical implementation based on the FFT and split Bregman algorithm. We draw the following conclusions about the advantages and disadvantages of these denoising models.

Among all the models compared, the first order TV and second order AC models are the best at preserving edges of objects in the piecewise constant image. However, TV causes image contrast loss while AC does not. TGV produces the best denoising results for the piecewise smooth image, and is able to remove the staircase effect associated with

the TV regulariser. However, TGV is the most computationally expensive approach. We also notice that the higher order models that involve the TV regulariser (i.e. TVL and TVBH) suffer from the staircase artefact, and applying directly the Laplace regulariser (i.e. AL model) or bounded Hessian regulariser (i.e. BH model) cannot achieve desirable denoising result. However, combining them with the TV regulariser (i.e. TVL and TVBH) improves the quality of denoising results. Moreover, higher order models that use the bounded Hessian regulariser (i.e. BH and TVBH) outperform their counterparts that use the Laplace regulariser (i.e. AL and TVL) in terms of both denoising results and computational efficiency.

With the knowledge of these models for image denoising, we can then use them appropriately for the different image segmentation tasks in the following chapters.

Chapter 3

Automated detection and quantification of RBC and histology images

In this chapter, we describe methods for segmenting red blood cell (RBC) images and histology images of microglial cells for diagnosis of diseases, namely, malaria parasite detection and microglia quantification, respectively.

3.1 Rapid malaria parasite detection and quantification

Malaria, one of the most devastating infectious diseases in the world, is caused by protozoan parasites of the genus Plasmodium. There are five major species of plasmodia that infect humans, out of which Plasmodium falciparum causes the majority of morbidity and mortality in Africa followed by less lethal Plasmodium vivax infections across South-East Asia, altogether infecting 200 million people and resulting in over half a million deaths every year [30]. Although drugs have been widely used to prevent and cure parasite-related diseases, parasites have acquired resistance to most of drugs both in Africa and South-East Asia, rendering them inefficient for future usage [73–75].

Early diagnosis and treatment are required to avoid anemia, organ failure [31] and deaths [76]. The lack of reliable methods and tools for use in field settings impedes early diagnosis in malaria endemic areas such as Africa. Traditional and widely practised method for malaria diagnosis relies on observable clinical symptoms, which are results of general host response to an infection. These methods demand expensive laboratory

facilities and highly trained personnel to conduct complicated analytical procedures and data analyses, which may not be feasible in malaria endemic countries.

The current gold standard for detecting malaria is microscopic examination of Giemsa-stained thin and thick blood smears [77] both in the field as well as in laboratory. This method allows detection of densely stained parasites against a background of lightly stained red blood cells (RBCs) and widely accepted owing to cost effectiveness, simplicity and rapidity. Though this method detects parasitemia levels of up to 1 infected cell in 10^6 cells [78], the microscopic examination requiring traditional bulky microscope is laborious and often fails when the parasitemia is low, a situation very common in the case of Plasmodium vivax infections. In addition, logistic issues and challenges associated with transporting the traditional bulky microscopes to remote and rural malaria endemic regions [79] and fulfilling operational and maintenance requirements could be challenging. On the other hand, manual counting from Giemsa-smears is known to vary depending on the personnel engaged and quality of smears. It can misinterpret other microorganisms (e.g., bacteria and fungi) as Plasmodium parasites and has difficulties identifying different Plasmodium strains [80]. Previously, several laboratories explored image processing based automated cell counting for parasitemia estimation, but most of them suffered from high false positive values [81–83], inferior accuracy [84], inability to differentiate parasitic stages [85] and the requirement of fluorescent dyes [86, 87].

To address these problems, our collaborators at the Singapore University Technology and Design have developed a low-cost, portable image-based cytometer that allows to image malaria parasite infected red blood cells (iRBCs). While in the University of Nottingham, we have tailored an image analysis framework for quantifying and classifying iRBCs imaged by the image-based cytometer. The proposed image analysis framework detects parasitic infections at levels as low as 0.2% and reliably classifies all parasitic life stages at different scales. We show the effectiveness of the proposed framework by comparing our results with those obtained by manual counting as well as by the commercial flow cytometer. In the following sections, we shall detail our image analysis framework.

3.1.1 Image acquisition

Blood used to simulate malaria parasite infection was purchased from Interstate Blood Bank. Before culturing malaria parasites, blood was transferred to EDTA Tubes (VACUETTE EDTA Tubes, Greiner Bio−One), washed three times in RPMI 1640 (Sigma−Aldrich) by centrifuging at 600g for 10 minutes to remove the buffy coat. RBCs were stored at 50% hematocrit in malaria culture medium (MCM), which is RPMI sup-

plemented with bovine serum (Albumax II, Gibco−Singapore), 2.5 $\mu g/mL$ gentamycin and hypoxanthine. Standard laboratory strains of Plasmodium falciparum, 3D7, were used in all experiments. Parasites were cultured in human O^+ erythrocytes at 2.5% haematocrit in MCM under standard conditions [88]. Parasites were synchronized by magnetic selection at late stages (46−48h) with the help of a SuperMACS magnet (Miltenyi Biotech, Bergisch Gladbach, Germany) and introduced into fresh RBCs, followed by sorbitol synchronization three hours later to allow a tight window of invasion [89].

A low-cost, portable image-based cytometer was then built for image acquisition from Giemsa-stained smears on the infected blood. As illustrated in Figure 3.1, it consists of a fine-focus (Z axis) adjustment platform (Dino-Lite, Model RK-10 Rack, Singapore), a 14 MP colour camera (ToupTek, P/N:TP114000A, Hangzhou, China), 20X objective or 100X oil immersion objective, two-dimensional translation stage and a white light source with tunable light intensity. Giemsa-stained smears were prepared carefully to obtain uniform cell distribution and placed on the 3D-printed sample holder for imaging. Images were captured with a commercial software (ToupTek, Hangzhou, China) by manually translating the smear slide on the stage. Only fields containing obvious debris were discarded. A 20X objective was used to capture large fields for imaging at high-throughput. However, the use of 20X objective is not sufficient to identify ring stage. In this case, a 100X oil immersion objective was utilized for ring forms imaging and parasitemia analysis.

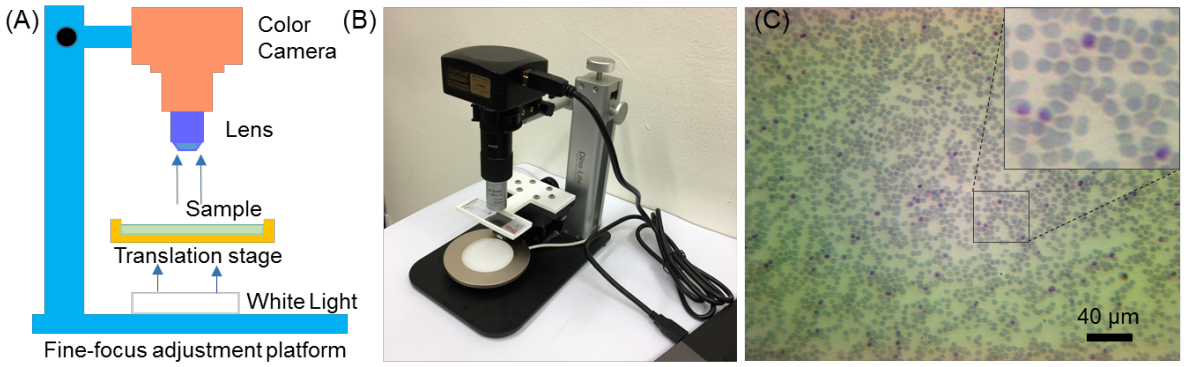


Figure 3.1: The image-based cytometer used to acquire RBC images. (A) Schematic experimental setup of the cytometer. (B) Photograph of the image-based cytometer. (C) An image of the blood smear taken by the cytometer, where the purple dots represent the RBCs infected by malaria parasites.

3.1.2 Image analysis framework

We used image processing and machine learning algorithms to analyse the smear images captured by the image-based cytometer. The image analysis algorithm consists of three key steps. First, a pre-processing total generalised variation (TGV) denoising method was used to remove the unavoidable noise in the original images. Second, an efficient and robust local adaptive thresholding approach was used to segment the smoothed image with possible intensity inhomogeneity. Finally, a machine learning method was performed to prevent false detections when the image does not contain any parasites. Details of each step will be discussed in the following.

The denoising of the original image was implemented by the TGV method [53], which is a powerful image pre-processing tool that has been extensively used in image processing community [66, 90, 91]. The TGV regularisation is capable of representing image characteristics up to an arbitrary order of differentiation (piecewise constant, piecewise affine, piecewise quadratic etc.). As detailed in Chapter 2, among all the compared methods TGV is the best higher order variational model for denoising a piecewise smooth image. The TGV model is of the following form

$$E\left(u,p\right)=\frac{1}{2\lambda}\int_{\Omega}\left(u-f\right)^{2}+\int_{\Omega}\left|\nabla u-p\right|+\alpha\int_{\Omega}\left|\varepsilon\left(p\right)\right|, \tag{3.1}$$

where $E(u,p)$ means the energy functional with respect to two variables u and p, u is the denoised/smoothed image, $p=(p1\ p2)$ is a symmetrised gradient vector which is closely related to $\nabla u=(\partial_x u\ \partial_y u)$, f is the input original image with intrinsic noise, and the operator $\epsilon(p)$ is the symmetrised derivative which is defined as $0.5(\nabla p+\nabla p^T)$. The first energy term on the right hand side of (3.1) is the data fidelity term which constrains the smoothed image u to be similar to the original image f. The second term preserves edges of objects in u and removes the noise from f in the meantime. The third term imposes smoothness on u and also eliminates the staircase artefact produced by the second term. The minimum of (3.1) is taken over all the symmetrised gradient of the deformation field $p=(p1\ p2)$ on the image space Ω. The positive λ coefficients and α in (3.1) balance the data fidelity term (the first energy term), the first order derivative (the second energy term) and the second order derivative (the third energy term). The value of α is normally set to 2 and this setting is suitable for most applications [91] and does not need to be tuned. In our experiment, we therefore set $\alpha=2$ and only vary λ to achieve different smoothness scales.

As TGV takes both first and second derivatives into consideration, it can highlight the edges of objects and smooth image without creating additional artefacts. This property

also enables TGV to overcome the problem of intensity inhomogeneity that has been found to widely exist in our captured images. Moreover, since the TGV functional (3.1) is convex, it guarantees a global optimal solution as well as allows the use of powerful modern optimization techniques. Finally, TGV is translationally invariant and rotationally invariant, meaning that the denoising results are not affected by the viewpoint of the images taken from different angles. In spite of the outstanding performance of TGV for image denoising, it is difficult to minimise such functional due to its nonlinear and nonsmooth nature. In this chapter, we minimise TGV with a fast numerical algorithm based on the split Bregman algorithm [21]. For a complete implementation on this algorithm, please refer to Chapter 2.

The smoothed image can be then segmented via a simple local adaptive thresholding approach, which is given as

$$p = \begin{cases} 0 & ls\,(I, ws) - I > C \\ 1 & \text{otherwise} \end{cases}, \tag{3.2}$$

where p is the binary image, $I = u$ in this case, and $ls(I, ws)$ means that I is convolved with a suitable operator, i.e. the mean or median filter. ws is the window size of the filter and C is a user-defined threshold value. The adaptive thresholding produces binary segmentation with fewer isolated points, giving a better result than a simple high-pass threshold. It is worth mentioning that this approach can segment large sized images (e.g. 2000×2000 pixels) with real-time computational speed, making it an ideal segmentation tool for the images used in the experiments. In addition, previous studies have shown that it is robust against inhomogeneity in medical images and can also obtain higher accuracy [92, 93].

To prevent false alarm for images that do not contain any parasites, we used machine-learning algorithms to automatically classify the parasite and non-parasite images. Specifically, 50 images were first selected manually as a training dataset, and divided into two groups, one containing 25 parasite images and the other group with 25 non-parasite images. Some of RBC images used for training purpose are shown in Figure 3.2 for visualisation. These 50 representative images were then segmented separately. For each RGB channel of the segmented image, the average intensity of all the pixels that fall in the segmented cells was calculated, thus forming a three-dimensional feature that can be effectively used for training a precise classification model. In this chapter, we choose Support Vector Machine (SVM) classifier. For the kernel function used in SVM, we tested the linear, quadratic and Gaussian radial basis kernel functions, as shown in Figure 3.3. We found that they led to comparable classification results. The simpler

linear kernel is finally used for SVM to train these extracted features. SVM classifies two different datasets by finding an optimal hyperplane that has the largest margin distance between them. It is more robust and accurate than other machine learning techniques. We therefore use the trained model by SVM to automatically classify the rest of unseen parasite and non-parasite images. The overall classification accuracy is around 95% on a unseen dataset of 140 RBC images.

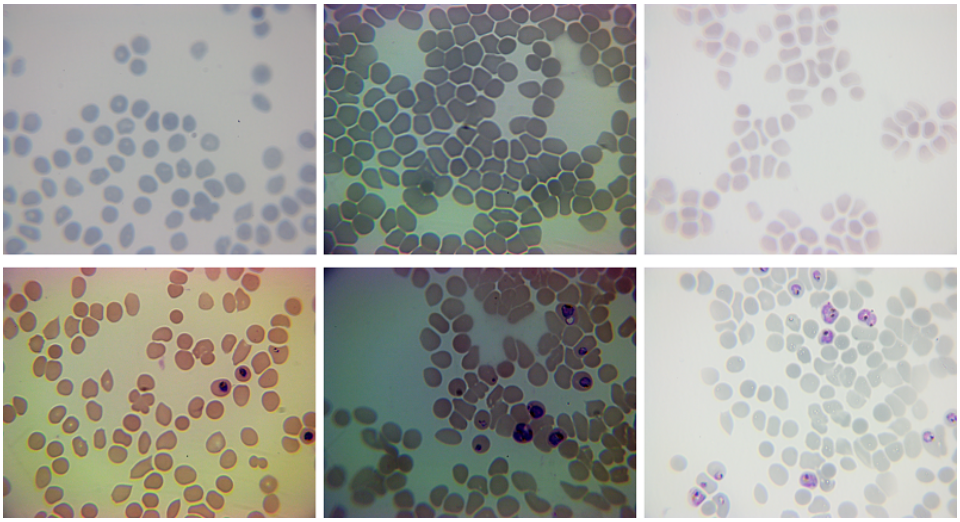


Figure 3.2: Selected RBC images. Row 1 and 2 show the RBC images without and with parasites, respectively.

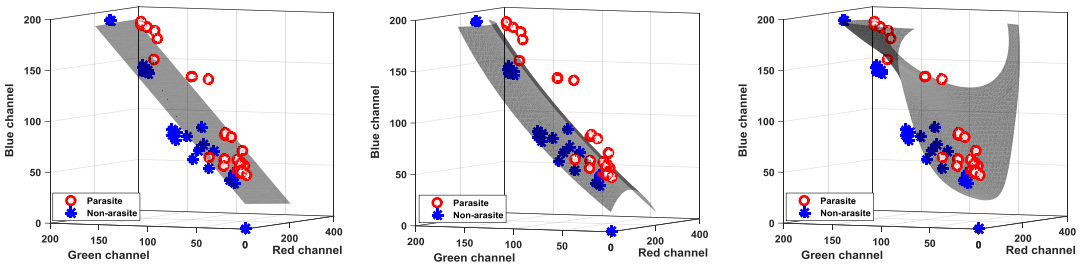


Figure 3.3: SVM classification. From left to right: classification results using SVM with the linear, quadratic and Gaussian radial basis kernel functions, respectively. Red and blue dots respectively represent the average RGB intensity of pixels in the parasite and non-parasite cells. 3D Gray plans are the optimal hyperplane that divides the training samples into two groups.

3.1.3 Quantifying iRBCs

After performing the proposed image analysis framework on the infected RBC images, we are able to quantify and classify the final segmentation results. As shown in Figure 3.4,

the flowchart of the quantification and classification is composed of three sections: 1) to estimate the total number of all RBCs; 2) to estimate the number of iRBCs; and 3) to classify distinct developmental stages of the parasite. For images acquired under objective 20X, we used the mean filter with the window size $ws = 120$ and set $C = 0.03$, $\lambda = 3$ for the segmentation of all RBCs, which produces binary segmentation with isolated RBCs and clustered RBCs. The number of RBCs in the clustered region was estimated by dividing the whole clustered area by the average area of a single RBC. For iRBCs segmentation, a high threshold value $C = 0.2$ was used with $ws = 100$, $\lambda = 3$. Similarly, under 100X oil immersion objective, $ws = 300$, $C = 0.01$ and $\lambda = 5$ were used for RBCs segmentation and $ws = 300$, $C = 0.15$ and $\lambda = 10$ were set for iRBCs segmentation. It should be noted that the C value can be affected by the light intensity. After standardisations, we kept a constant, optimum light intensity and fixed exposure time of camera for our further assays. To address the possible risk of counting multiply infected parasites in one single iRBC as multiple iRBCs, we performed the following calculation: if the distance between two segmented parasites is less than 150 pixels, these parasites will be considered as single iRBCs. It might still cause the false classification when two overlapping iRBCs are observed, but this was an extremely rare case due to the careful preparation of thin and uniform smear slides.

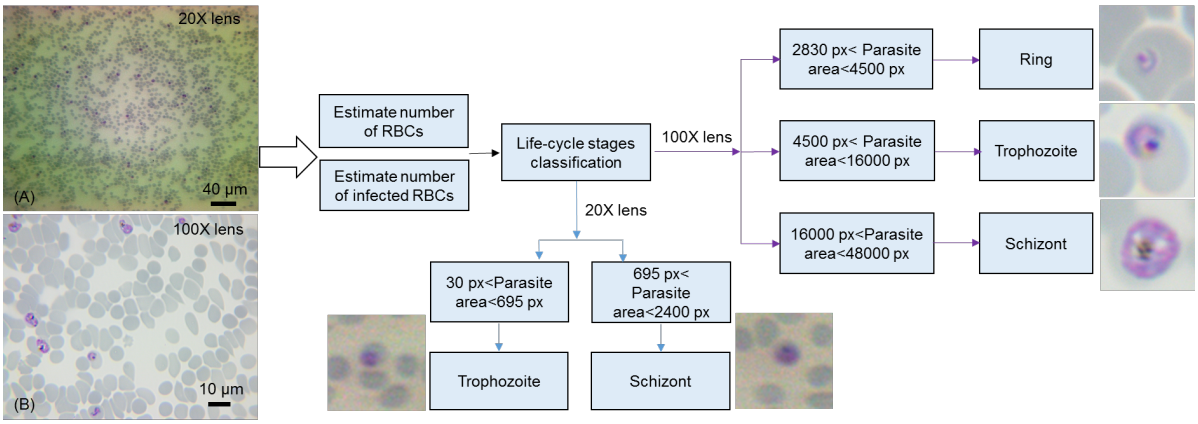


Figure 3.4: Flowchart of classification of parasites' developmental stages. Image A was taken by objective 20X and image B was taken by oil immersion objective 100X.

3.1.4 Classification of parasites' developmental stages

After estimating the number of iRBCs, we performed the stage classification based on the area occupied by individual parasites within an iRBC. Figure 3.4 depicts the criteria of parasites' stage classification under 20X and 100X magnifications. For 100X oil

immersion objective, if the area occupied by the parasite is larger than 2830 pixels and less than 4500 pixels, then it was identified as a ring-stage parasite. It was classified as trophozoite stage when the area is greater than 4500 pixels and less than 16000 pixels. If the parasite-occupied area is greater than 16000 pixels and less than 48000 pixels, it was considered as a schizont. For 20X objective, the magnification was insufficient to image the rings, thus we only identify the two late stages under this low magnification. All these threshold pixel numbers for stage classification were verified independently based on conventional cytological examination. The developed system was used to determine parasitemia from seven independent samples. Each sample set included three smears and a minimum number of 30 fields were captured continuously resulting in counting at least 4000 RBCs for each smear. Imaging 30 fields for each smear took about 10 minutes, meaning it took 20 seconds to image each field.

3.1.5 Results and discussion

Image analysis results under 20X objective. The 20X objective was used for high-throughput imaging, and the analysis of images captured by the image-based cytometer were performed automatically using our image analysis algorithm, as shown in Figure 3.5. Table 3.1 shows the results from six randomly selected images from the same smear by using the image analysis algorithm and manual counting. Errors of total RBCs, iRBCs and parasitic classification were less than 5%. These negligible errors of total RBC number were primarily in regions where RBCs were clustered, minimally compromising the counting accuracy. Estimation of the number of iRBCs and life cycle stages were mildly influenced by debris. Nevertheless, error values less than 5% showed the excellent performance and reliability of the proposed image analysis framework compared to manual counting.

Image analysis results under 100X objective. Since 20X objective is insufficient to image the ring stages with high accuracy, a 100X oil immersion objective was used for malaria parasite detection and classification. As illustrated in Figure 3.6, image-based cytometer equipped with 100X allowed us to reliably classify all the three stages of parasites. Table 3.2 shows the inspection results of RBCs, iRBCs segmentation and stage-specific classification. The ability of our proposed image analysis framework to classify and identify different developmental stages may lead to potential pharmacological applications in identifying the potential blockers of egress, invasion and parasitic development inhibitory molecules [86]. In a single image, errors of a specific parasite stage may be large (due to limited number of parasites in the field), but the total error of randomly selected 6 images was well below 5%, which shows the robustness of our system.

Giemsa-stained images taken at 20X and 100X objective lens for the aforementioned experiments are respectively shown in S1 and S2 figures in the supplementary of [94].

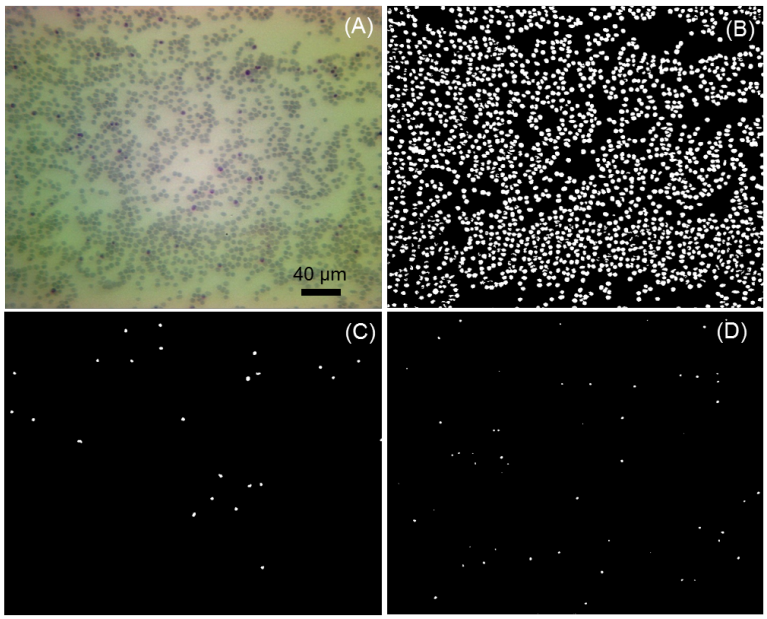


Figure 3.5: Images captured by the cytometer in Figure 3.1 are automatically analysed by the developed image analysis framework. (A) Original image taken by objective 20X. (B) Representation of all the extracted RBCs. (C) Extracted schizont stage iRBCs. (D) Extracted trophozoite stage iRBCs.

Table 3.1: Comparison of cell counting and developmental stage classification (Objective 20X). MC represents the average value of two manual counting performed in the same image which was analysed by the image-based cytometer. iRBCs - infected Red Blood Cells. IBC indicates the result analysed using the proposed image analysis framework within the image-based cytometer.

Image No.	1	2	3	4	5	6	Total
MC (RBCs)	1933	2381	2788	2870	2821	1942	14735
IBC (RBCs)	2000	2340	2810	2753	2801	1972	14676
Error (RBCs)	3.35%	1.72%	0.79%	4.10%	0.71%	1.55%	4.00%
MC (iRBCs)	103	90	127	131	134	78	663
IBC (iRBCs)	106	94	126	131	137	78	672
Error (iRBCs)	2.91%	4.44%	0.78%	0%	2.24%	0%	1.36%
MC (Schizont)	50	36	70	63	71	25	315
IBC (Schizont)	51	35	72	64	72	25	319
Error (Schizont)	2.00%	2.78%	2.86%	1.59%	1.40%	0%	1.27%
MC (Trophozoite)	53	54	57	68	63	53	348
IBC (Trophozoite)	55	59	54	67	65	53	353
Error (Trophozoite)	3.77%	9.26%	5.26%	1.47%	3.17%	0%	1.44%

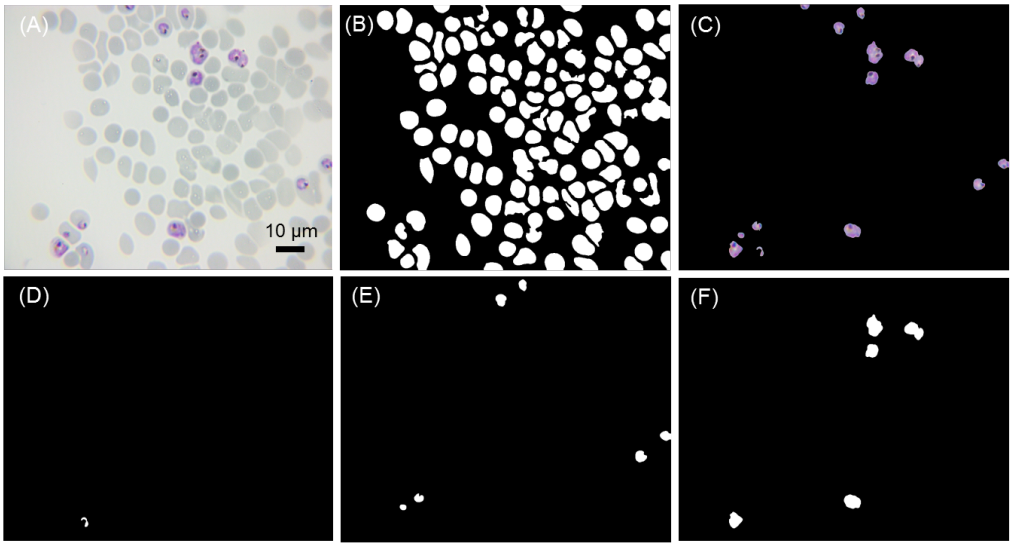


Figure 3.6: Images captured by the cytometer in Figure 3.1 are automatically analysed by the developed image analysis framework. (A) Original image taken by immersion objective 100X. (B) All the extracted RBCs. (C) All the extracted iRBCs. (D) Extracted ring stage. (E) Extracted trophozoite stage. (F) Extracted schizont stage.

Table 3.2: Comparison of cell counting and developmental stage classification (Immersion objective 100X). MC represents the average value of two manual counting performed in the same image which was analysed by image-based cytometer. iRBCs - infected Red Blood Cells. IBC indicates the result analysed using the proposed image analysis framework within the image-based cytometer. The error rate is not computed if the number of iRBCs is below 20.

Image No.	1	2	3	4	5	6	Total
MC (RBCs)	189	132	170	182	122	160	955
IBC (RBCs)	183	133	162	184	123	165	950
Error (RBCs)	3.17%	0.76%	4.71%	1.10%	0.82%	3.13%	0.52%
MC (iRBCs)	10	13	9	8	9	7	56
IBC (iRBCs)	10	13	8	8	9	7	55
Error (iRBCs)	–	–	–	–	–	–	1.79%
MC (Schizont)	6	5	5	4	5	4	29
IBC (Schizont)	6	5	4	4	6	4	29
Error (Schizont)	–	–	–	–	–	–	0%
MC (Trophozoite)	4	7	4	4	3	1	23
IBC (Trophozoite)	4	7	4	3	2	2	22
Error (Trophozoite)	–	–	–	–	–	–	4.35%
MC (Ring)	0	1	0	0	1	2	4
IBC (Ring)	0	1	0	1	1	1	4
Error (Ring)	–	–	–	–	–	–	–

We evaluated the capability of our image analysis framework over a range of parasites and the results were compared against those obtained by the commercial flow cytometry.

Mixed stage parasites were diluted to obtain parasitemia (the demonstrable presence of parasites in the blood) ranging between 0.2% and 9%. Giemsa-stained smears were taken for assessing the parasitemia (in a blind manner to not reveal sample identity) using image-based cytometer and flow cytometry-based. As shown in Figure 3.7, parasitemia obtained using the image-based cytometer with our image analysis framework is totally comparable to those obtained from the flow cytometric experiments as confirmed through two-tailed paired T test.

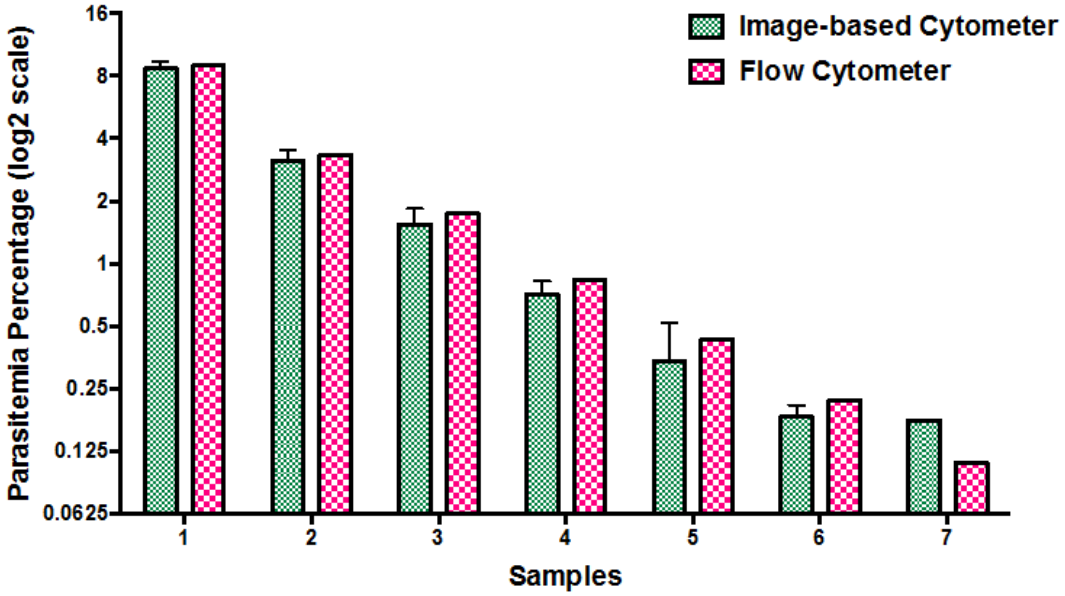


Figure 3.7: Comparison of parasitemia values using image-based cytometer (with the proposed image analysis framework) and flow cytometer. The results were comparable with flow cytometer values with no significant difference. Each sample has 20+ RBC images.

In summary, we have proposed a novel image analysis framework for scoring and staging malaria parasites for an image-based cytometer. By comparing the results obtained by our image analysis framework to those of manual counting and flow cytometry, we have validated its robustness and accuracy for scoring and staging malaria parasites. The image analysis framework enpowers the cytometer, making it a fast, reliable and affordable tool for detecting and screening malaria, providing comparable results to commercialised flow cytometry.

3.2 Microglia segmentation and quantification

Microglia are immune cells exclusive to the central nervous system (CNS) and about 1.5 trillion of them reside in the brain and spinal cord [32, 95]. In response to a variety of signals, microglia show a range of phenotypes, from protective to detrimental associated with motility and morphological changes [96]. In the healthy brain, microglia constantly survey the surrounding tissue with extended processes, clear debris from dead cells, and prune and maintain brain synapses. They are also essential to learning and memory [97, 98], protect neurons from damage, and mediate pain [99, 100]. In response to an injury or infection, microglia initiate an early, protective response by moving towards the site of injury, where they release a cascade of chemicals leading to repair of the damaged area. Microglial activation is a hallmark of chronic neuroinflammation, which is believed to play an important role in a range of brain disorders, which has yet to be fully understood, including stroke, multiple sclerosis, Parkinson's, Huntington's and Alzheimer's disease [34–36], and can also reflect a neuroprotective behaviour in these chronic conditions [96].

The heterogeneity of microglial functions is in part linked to their shape and activation state, and much information can be obtained from their morphological characteristics [101]. Microglial cell shape evolves from a resting fully ramified shape with extending processes and smaller soma, to the fully activated amoeboid shape with a larger soma and shorter processes [102, 103]. To date, microglia activation has been linked to three distinct functions: a classical pro-inflammatory activation state, an alternative activated anti-inflammatory state and a complementary deactivation state associated with an anti-inflammatory and functional repair phenotype [96]. Classifying microglia activation states in histological images can help pathologists with disease diagnosis [104], provides key information for understanding diseases of the central nervous system [105] and is essential for the validation of in vivo biomarkers that allow the stratification and monitoring of patients and populations at risk [1], [106].

In order to study microglia quantitatively and classify microglia activation states in a histology image, the first step is normally to extract these microglia such that the number, size and shape of microglia can be analysed subsequently within regions of interest. Manual or semi-automated segmentation methods are time-consuming and require user intervention [37, 38] with an element of subjectivity and inter-observer variability. Developing an automated microglial segmentation method is thus curial for such an analysis task. Image analysis approaches commonly used for quantifying histology images that rely on thresholding struggle with intensity inhomogeneity within the images. In this chapter, new methods are developed for quantitative analysis of microglial images.

Microglia are segmented using the Mumford-Shah total variation [107, 15, 108, 68, 109] and the fast split Bregman [21, 25, 24, 66] methods. We also show that this framework offers advantages over manual analysis of histology data of wild type mice and transgenic mouse models of Alzheimer's disease. The proposed segmentation method is fast, robust and accurate and is scalable to large datasets, allowing the quantitative analysis of microglia in regions of interest as well as across the whole brain.

3.2.1 Development of segmentation method

One of the major tasks of microglia analysis is to calculate the sizes of microglial bodies and processes. As such a weak smoothing (α=10) and a strong smoothing (α=300) are applied to the grayscale image for segmenting microglia process and soma respectively.

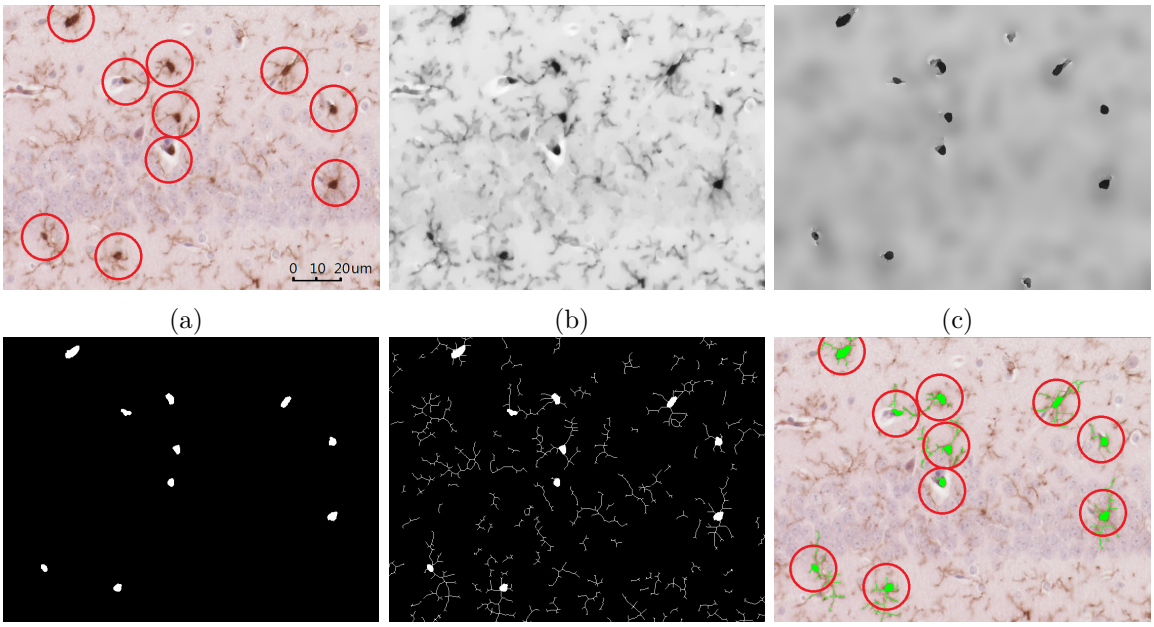


Figure 3.8: Workflow of the proposed segmentation for a sample image. (a) Original histology image. (b) Smoothed image (α=10). (c) Smoothed image (α=300). (d) Soma segmentation. (e) Soma and processes segmentation. (f) Automatically labelled microglia overlaid onto the histology image. Inclusion criteria: soma size larger than 16.7μm.

Noise is inherent in histology images. Research for quantitative analysis of microglial often relies on thresholding (manual or automatically) [110–112]. These methods are not suitable for dealing with noisy and inhomogeneous histological images of microglia. A preprocessing step is therefore needed to remove noise/outlines whilst preserving the details of microglia in the image. An adaptive thresholding algorithm (3.2), which

automatically determines threshold values for different parts of the image, is then applied to the denoised images to extract the microglia. Tiny microglia with a soma size smaller than 16.7μm are also removed, as suggested in [110]. In Alzheimer's disease, clusters of microglia with the morphological appearance of an activated phenotype are found around amyloid plaques [113]. These clusters are detected by their abnormal soma sizes and analysed separately. Finally, the segmented microglia processes are skeletonised and combined with the segmented microglia soma, and the isolated microglial processes, not connected to any microglia soma, are removed. Figure 3.8 shows the workflow of the proposed segmentation method.

3.2.2 Denoising with Mumford-Shah total variation segmentation

If we look at the histology image in Figure 3.8 (a), we can see that there exist unfavourable outlines outside the regions highlighted in the red circles. Their intensity values are very closed to the microglia in the circles, while the scale of these outlines are much smaller than that of the microglia. The idea here is to segment the image and only get the microglia, so we intend to use a filter to smooth the outlines out while remain the microglia we intend to segment. Ideally one would like such a smoothing method which does not destroy the edge information. This dilemma between smoothing noise and preserving edges and corners has been commonly considered as a "chicken and egg" problem: An object of interest is more easily segmented, if one smooths the grey value across the area corresponding to the object; however, in order not to smooth across the boundaries of the object, one already needs to know where the object is.

Interestingly, this "chicken and egg" dilemma can be tackled by a variational approach. Mumford and Shah [107] proposed to approximate a given input image f with a piecewise smooth function u by minimising the functional

$$E(u, C) = \frac{1}{2}\int_{\Omega}(f-u)^2 dx + \frac{\alpha}{2}\int_{\Omega/C}|\nabla u|^2 dx + \beta\,|C| \tag{3.3}$$

simultaneously with respect to the image u and with respect to the contour C. The first term is a fidelity term, which enforces that the function u is similar to the input image f in the L2 sense. The second term enforces smoothness of the segmented image but permits discontinuities of u across a boundary denoted by C. The last term gives the 1D Hausdorff measure of the length of the boundary. The parameter α defines the spatial scale on which smoothing is done. β measures the length of the object boundary.

This free discontinuity problem in (3.3) triggered a large number of detailed studies on the region-based segmentation. However, it is very difficult to minimise the non-convex Mumford-Shah functional (3.3) as it includes energy terms that are defined in spaces of different dimensions. In order to implement the Mumford-Shah model mathematically, it is necessary to unify the energy terms in the model to a space of same dimension. There are two numerical approaches to do so. The first approach [114–117] applies the concept of shape derivatives (i.e., contour normal), transforming the energy terms in the model defined in 2D image space into the ones defined in 1D contour space. Consequently, the Mumford-Shah model becomes a parametric active contour model. The level set scheme [118] is then integrated into the parametric model for handling topological changes during contour evolution. The second approach involves the use of the characteristic functions for different phases or classes in an image (a phase/class contains pixels having similar characteristic), transforming the 1D terms into the 2D ones. Example methods of this approach include: 1) the variational level set method [119–123, 65] that combines the classical level set method with the variational method; 2) the variational label function method, also known as the piecewise constant level set method [124–126] or the fuzzy membership function method [127, 128]; 3) the Γ-convergence elliptic function approximated method [129, 108, 109, 68].

The main advantage of the Γ-convergence approximated method over the others is that it can denoise and segment a piecewise smooth image simultaneously. In this section, we will therefore focus on this method. Ambrosio and Tortiroilli [129] first implemented the original Mumford-Shah functional by using the Γ-convergence elliptic function approximation. Their model is given by

$$E_{\varepsilon}(u,v)=\int_{\Omega}(u-f)^{2}dx+\alpha\int_{\Omega}v^{2}\left|\nabla u\right|^{2}dx+\beta\int_{\Omega}\left(\varepsilon|\nabla v|^{2}+\frac{(v-1)^{2}}{4\varepsilon}\right)dx. \tag{3.4}$$

The regularisation term of this model is defined in the quadratic L2 space (Tikhonov), which can over-smooth/blur image edges and lead to incomplete segmentation boundaries. A better edge preservation regulariser, the first order total variation [15], was thus intoduced by Shah [108] to replace the quadratic L2 regularisation term in (3.4). However, as the total variation favours piecewise constant solutions, Shah's model (named the Mumford-Shah total variation model in the chapter) suffers from the undesirable staircase artefact. However, as we can see, such staircase artefact would not appear since the histology images we are concerned about tend to be piecewise constant. Our previous research [68, 109, 130] has shown that Shah's model is fast and accurate and is therefore chosen to denoise the histology images here. It can smooth noise out and preserve the

edges of microglia, making it easier to detect them from histology images. Furthermore, the model can benefit from fast imaging solvers such as the FFT and shrinkage, which makes it very efficient to implement. The Mumford-Shah total variation model works as follows [108]

$$E_\varepsilon(u,v) = \int_\Omega (u-f)^2 dx + \alpha \int_\Omega v^2 |\nabla u| \, dx + \beta \int_\Omega \left(\varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon} \right) dx, \tag{3.5}$$

where α, β and ε are three positive parameters balancing the energy terms, u is a piecewise constant function to approximate the original image f, and v is a piecewise function to represent object edges in the image (v takes value 0 on the edges and 1 in smooth regions). This energy functional can be used to smooth microglia as well as find the edges of microglia in a histology image.

A fast split Bregman algorithm [21] is designed for discretising and solving (3.5). This algorithm has been widely used to solve L1-based variational models [65, 131, 64, 25, 24, 66], as introduced in Chapter 2. An auxiliary vector $w = (w_1 \; w_2)$ and a Bregman iteration parameter $b = (b_1 \; b_2)$ are introduced to transform the minimisation of (3.5) into optimising the energy functional as follows

$$\begin{aligned} E_\varepsilon(u,v,w) = & \int_\Omega (u-f)^2 dx + \alpha \int_\Omega v^2 |w| \, dx \\ & + \frac{\theta}{2} \int_\Omega (w - \nabla u - b)^2 dx \\ & + \beta \int_\Omega \left(\varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon} \right) dx, \end{aligned} \tag{3.6}$$

where θ is positive penalty parameter. In practice, each variable u, w and v in functional (3.6) is minimised separately. For example, the variables v and w are fixed first, and the Euler-Lagrange equation of u is as follows

$$u + \theta \Delta u = f + \theta div \left(w^k - b^k \right), \tag{3.7}$$

where div and Δ denote the divergence operator and Laplace operator respectively, and k stands for the current iteration. By applying the discrete Fourier transform to both sides of the equation, the closed-form solution of u is obtained as

$$u^{k+1} = \Re \left(\mathcal{F}^{-1} \left(\frac{\mathcal{F}(f) + \theta \mathcal{F}(div) \, \mathcal{F}\left(w^k - b^k \right)}{1 + \theta \mathcal{F}(\Delta)} \right) \right), \tag{3.8}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denotes the discrete Fourier transform and inverse Fourier transform respectively. $\Re(\cdot)$ is the real part of a complex number, "—" stands for pointwise division of matrices. The minimisation with respect to w can be expressed as follows

$$w^{k+1} = \underset{w}{\operatorname{argmin}} \left\{ E(w) = \alpha \int_{\Omega} v^2 |w| \, dx + \frac{\theta}{2} \int_{\Omega} (w - \nabla u - b)^2 dx \right\}. \tag{3.9}$$

It is easy to check that

$$w^{k+1} = \max\left(\left| \nabla u^{k+1} + b^k \right| - \frac{\alpha}{\theta} v^2, 0 \right) \frac{\nabla u^{k+1} + b^k}{|\nabla u^{k+1} + b^k|}, \tag{3.10}$$

with the convention $0/0 = 0$.

The above equation is known as the analytical soft thresholding equation or shrinkage. Note that the shrinkage (3.7) includes two subshrinkages for each component of the vector w. Next, the Euler-Lagrange equation of v with u and w fixed is obtained as follows

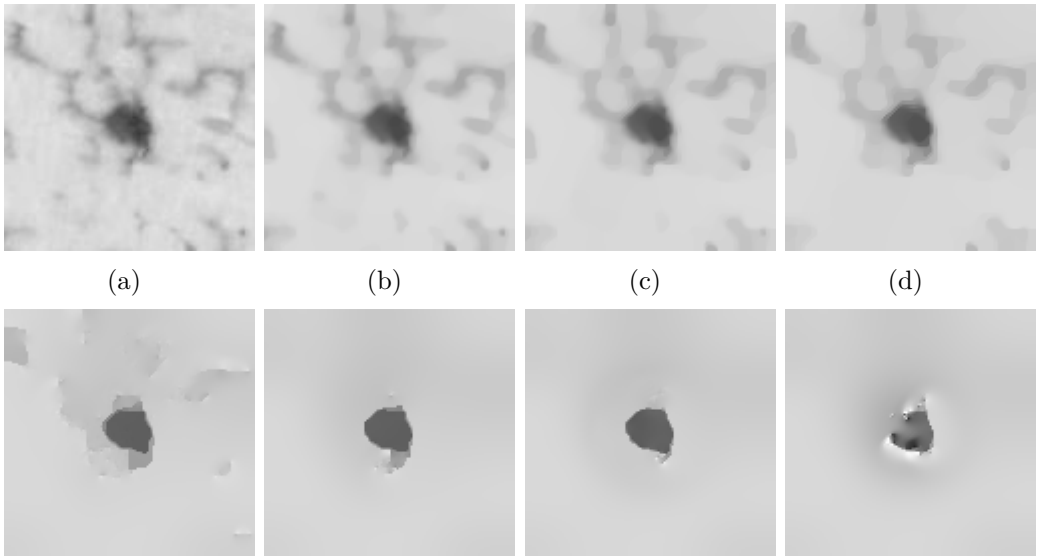
$$2\alpha v \left| w^{k+1} \right| - 2\beta\varepsilon\Delta v + \beta \frac{(v-1)}{2\varepsilon} = 0. \tag{3.11}$$

This equation can be efficiently solved approximately by one iteration of Gauss-Seidel. Finally, the Bregman parameter is updated using

$$b^{k+1} = b^k + \nabla u^{k+1} - w^{k+1}. \tag{3.12}$$

Note that the equations (3.8), (3.10), (3.11) and (3.12) are iteratively updated until a convergence criterion is met. The parameters α, θ, β and ε in (3.6) should be adjusted. α is a smoothing parameter and larger α gives smoother result. We set α=10 as a weak smooth and α=300 as a strong smooth. The selection of α was based on the results of a series of experiments using different smoothing values on microglial images. Figure 3.9 shows example results of a single microglial cell that was smoothed using different α values. It can be seen that the method produced the best results when α=10 and α=300 were chosen for, removing noise, and simultaneously preserving the details of microglial cell body and processes in the images respectively. As compared to the total variation denoising results in Figure 3.10, the Mumford-Shah total variation better preserves the shape edge of the cell, thus enabling more accurate segmentation results.

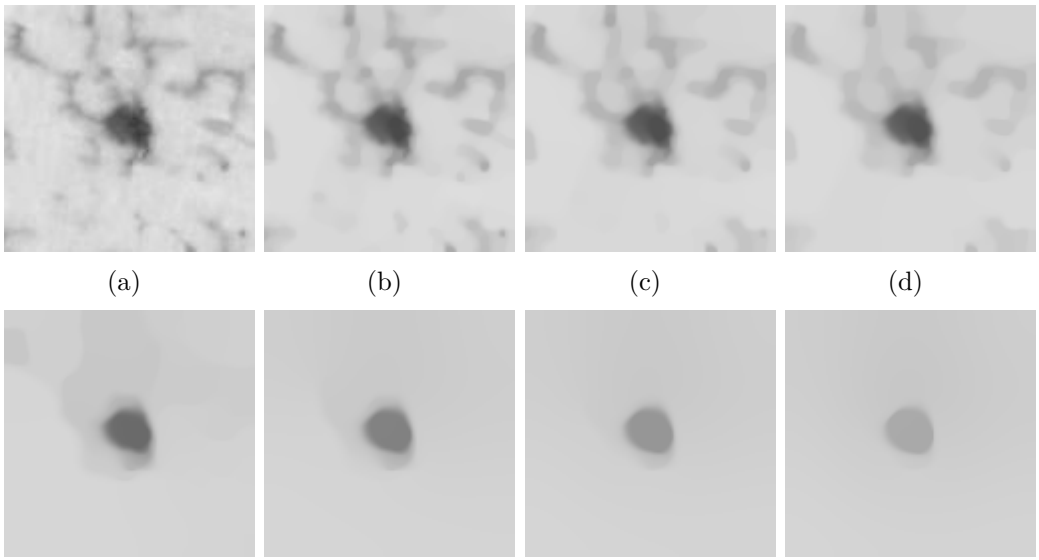
Due to the Bregman iteration technique used, different penalty parameter θ will provide similar smooth result. However, the algorithm may have different rate of convergence with different values of θ. In all the experiments, the value of θ is fixed



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

(e) (f) (g) (h)

Figure 3.9: Smoothed a single microglial cell using Mumford-Shah total variation. (a) α=0. (b) α=10. (c) α=20. (d) α=30. (e) α=100. (f) α=200. (g) α=300. (h) α=400.



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

(e) (f) (g) (h)

Figure 3.10: Smoothed a single microglial cell using total variation denoising. (a) α=0. (b) α=10. (c) α=20. (d) α=30. (e) α=100. (f) α=200. (g) α=300. (h) α=400.

as 5 in order to achieve a fast convergence rate. Parameter β balances the last energy term against the other three terms in model (3.6). It is empirically chosen as 0.1 for all experiments. The approximation of the Mumford-Shah regulariser term in the Mumford-

Shah total variation model (i.e., the last two energy terms in (3.5)) is based on the phase field theory under Γ-convergence [129]. Theoretically, the parameter ε should be close enough to zero to satisfy such approximation. Therefore, we set $\varepsilon = 0.0001$ for all the experiments.

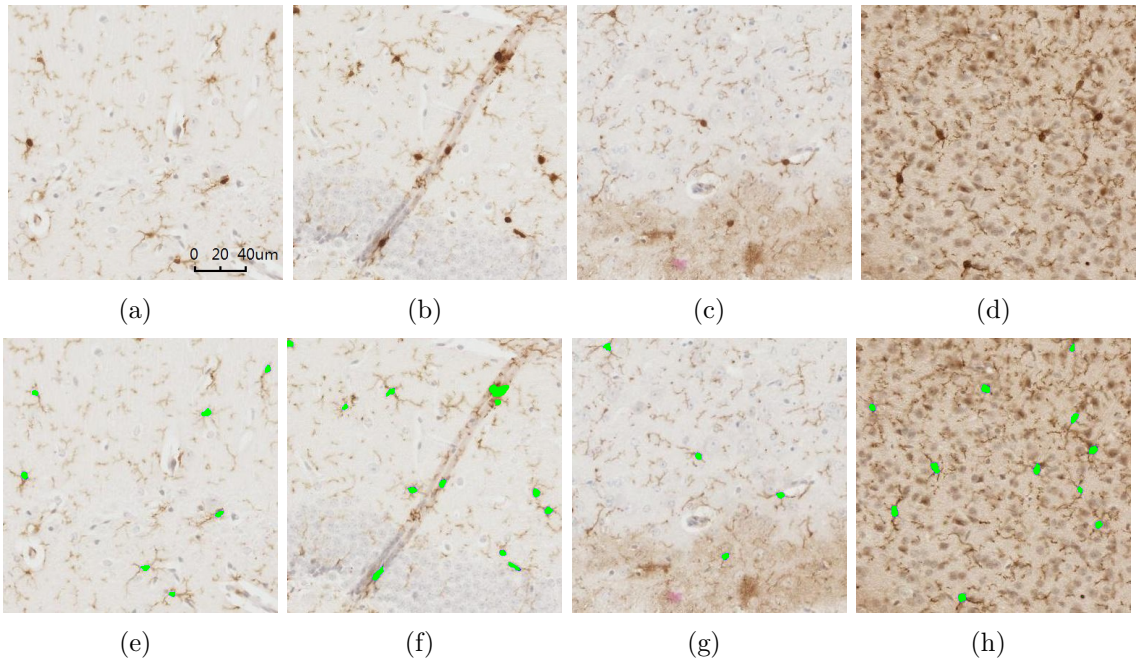


Figure 3.11: Challenges described in [1] and the proposed method overcomes these problems. (a) Image shows microglia cells with strong and weak intensities. (b) Image has a large complex artefact with microglia located partly inside the artefact. (c) image shows two regions that exhibit different visual textures. (d) Image displays a complex texture appearance that blurs the distinction between microglia cell and background pixels. (e)-(h) Microglia soma labelled using the proposed method.

3.2.3 Experimental data acquisition

The data used in this chapter were generated from brain tissue from female mice, transgenic APPswe/PS1dE9, a mouse model of Alzheimer's disease, or their wild-type littermate. All mice were were bred in the University of Nottingham's Biomedical Service Unit as previously described [132]. Some of these mice had been treated 10 days before with a lipopolysaccharide immune challenge (LPS, 100ug/kg) known to selectively activate microglia or its vehicle Phosphate Buffered Saline (PBS, Sigma Aldrich, St. Louis, MO, USA). The genotype and treatment condition ensured a wider representation of morphological states but were not analysed systematically as the focus on the chapter

is on classification. All procedures were approved as required under the UK Animals (Scientific Procedures) Act 1986. Brains were fixed in 4% Paraformaldehyde for at least 24 hours at 4°C and embedded in paraffin wax on a tissue embedding station (Leica TP1020). 7μm-thick coronal sections were cut throughout the hippocampus using a microtome, mounted on 3-Aminopropyltriethoxysilane-coated slides and dried overnight at 40°C. Immunostaining was carried out using standard procedures at room temperature, as described below.

All the solutions were freshly prepared using PBS + 1% Tween 80, except DAB solution that was prepared in distilled water. Briefly, the tissue was re-hydrated in consecutive rinses in Xylene, 100% ethanol, 70% ethanol and distilled water. Antigen retrieval was performed by 20 minutes incubation in Sodium Citrate buffer at 95-99°C, followed by incubation in 1% H2O2 (Sigma Aldrich, St. Louis, MO, USA). Tissue was then blocked in 5% normal goat serum (Vector Laboratories, Burlingame, CA), incubated in rabbit polyclonal anti-Iba-1 primary antibody (1:6000; WAKO Chemicals, VA, USA) for 1 hour followed by 30 minutes incubation with anti-rabbit secondary antibody (1:200; Vector Laboratories Inc. Burlingame, CA). After washing, sections were incubated with Vectastain Elite ABC kit (Vector Laboratories Inc. Burlingame, CA) and labelled with DAB peroxidase substrate (Vector Laboratories, Burlingame, CA) according to manufacturer's instructions. To reveal histologic morphology, sections were then lightly counterstained with haematoxylin (purplish-blue nuclear stain) and eosin (pink cytoplasmic stain) and mounted with DPX-mount media.

Digital focused photo-scanning images were acquired using a Hamamatsu NanoZoomer-XR with TDI camera technology at a magnification of 20X. Rectangular regions of interest (ROIs) were drawn within the hippocampus subfields with an area of 0.2 mm^2 or 0.1mm^2 using NDP.view2.

3.2.4 Validation of segmentation method

Previous studies have identified the limitations of the existing microlia segmentation methods [1]. This includes: microglia contrast issues within the same image, large artefacts, different visual textures within the same field of view and textures that blur the distinction between microglia cell and background. Experiments show that the proposed segmentation method is accurate and overcomes these problems. Some example segmentation results are shown in Figure 3.11.

20 images randomly selected from all experimental condition were used for validation of the proposed automated technique against manual analysis. For manual segmentation, the soma of microglial cells was delineated using the freehand line tool in NDP.view2 at

a magnification of 40X and the data (number of soma per ROI and soma size in μm^2) extracted. ROIs were exported 20X and saved as Jpeg images for subsequent analysis of the percentage of area stained, using ImageJ [39]. Images were split into red, green and blue using the RGB stack command, prior to thresholding. The blue stack was chosen to eliminate non-specific highlighting of the neuronal nuclei. The threshold level was adjusted manually for each image to highlight the soma and processes, as well as the percentage area stained. For details see Figure 3.12.

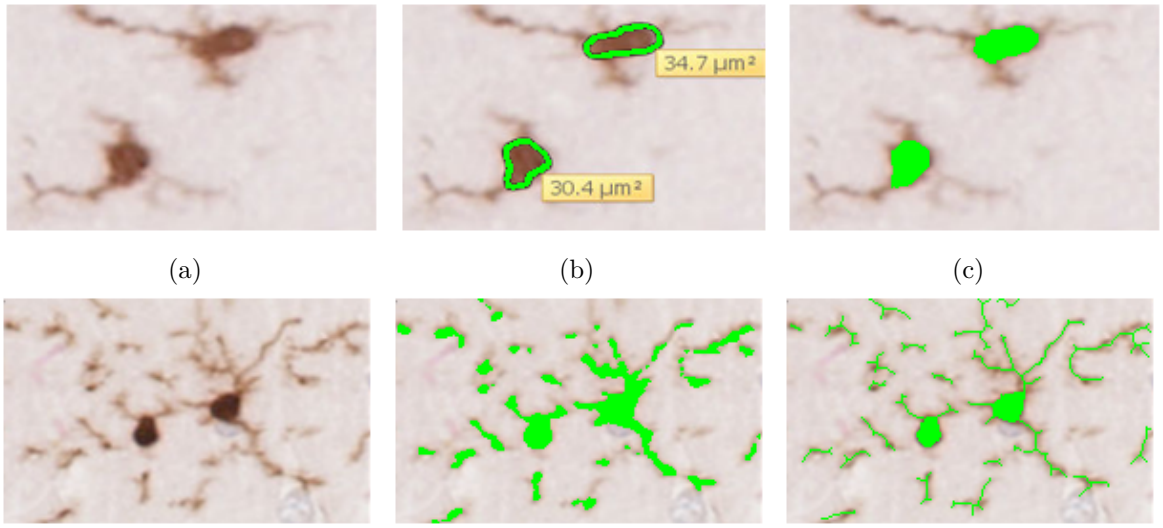


Figure 3.12: Examples of soma area estimation and estimation of the percentage of the area stained. (a) and (d) Iba-1 positive microglial cells in unprocessed images. (b) Freehand delineation of microglial somas and annotation of the soma size calculated by NDP.viewer2 software. (c) Soma segmentation by the proposed method. (e) Example of manually adjusted threshold level for the estimation of the percentage of area stained by Iba-1. (f) Example of automatic estimation of the percentage of area stained by Iba-1.

Example of an automated segmentation result by the proposed method for analysing the number of soma per ROIs and soma size in μm^2 is shown in Figure 3.12 (c). The automatic estimation of the percentage of area stained is shown in Figure 3.12 (f). The results by the proposed automatic method and the manual method on the image dataset are compared. As shown in Figure 3.13, the proposed method has produced similar results as those obtained by the manual method for the number of cells per ROI. From Figure 3.13 (a) and (b), one can see that the green solid areas segmented by our method better overlap the true soma in the image as compared to manual contours annotated by an experienced expert. Differences between scorers for the manual analysis are due to differences in judgement in defining the border of the soma. For percentage area

stained, the proposed method achieves expert desired results as compared to the manual thresholding method. Such a method is very dependent upon the scorer's judgement leading to inconsistencies between scorers, as shown in Figure 3.12 (e).

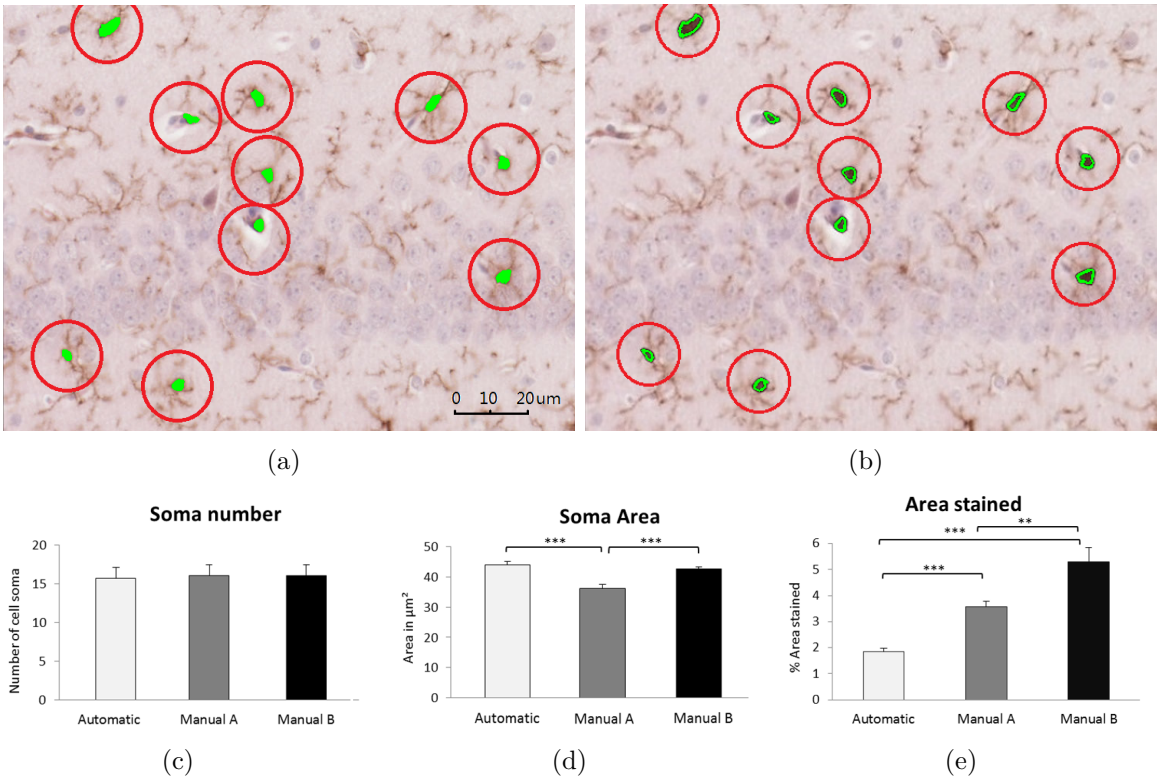
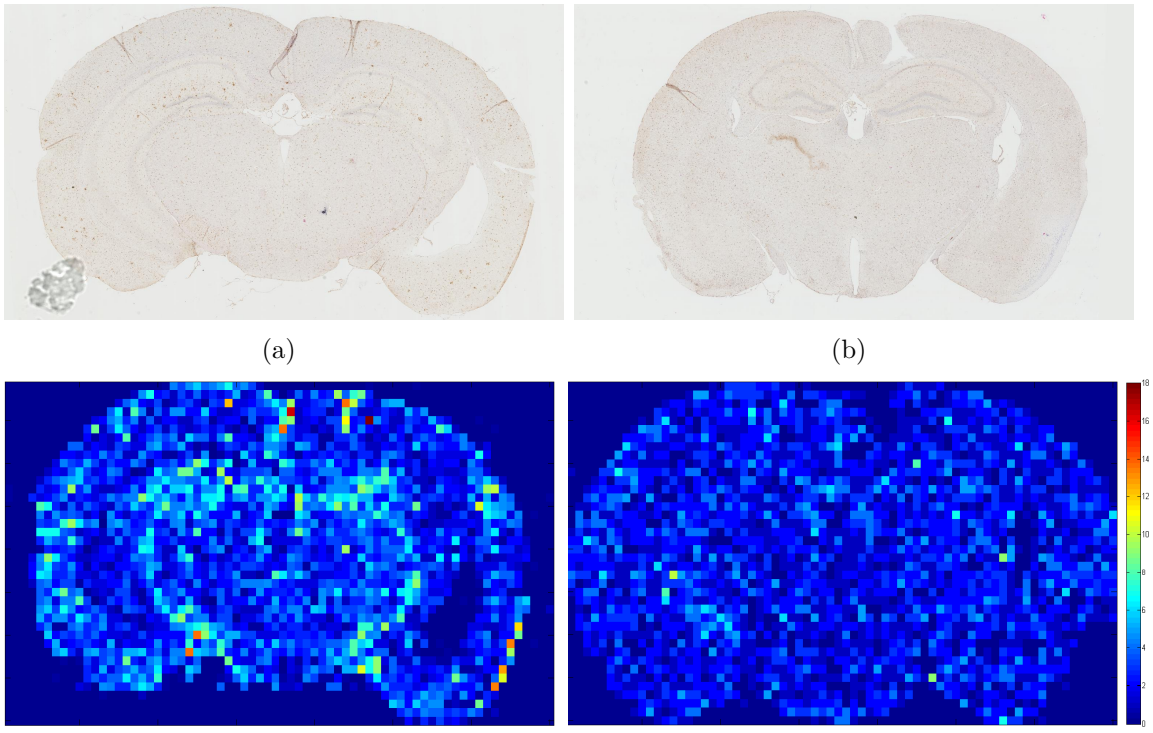


Figure 3.13: Comparison of segmentation results using the proposed segmentation method and the manual segmentation method. (a) Microglia soma segmented using the proposed automatic segmentation method. (b) Microglia soma manually segmented by an expert. Analysis results calculated using automatic and manual segmentation methods within the hippocampus. (c) Soma number (d) Soma area. (e) Percentage area stained. Automatic: results produced by the proposed method. Manual A and Manual B: results produced by the experts. Data are presented as means + standard error.

The scalability of the proposed methods on large datasets is tested in the analysis of microglial distribution in images of both healthy mouse brain and Alzheimer's Disease mouse brain models, each of which is 8640×15360 in size. Two examples of these images are shown in Figure 3.14 (a) and (b). Processing an image with this size is very time-consuming for most image processing algorithms. Our method is however fast, accurate and robust for segmenting microglia cells from the whole brain in a histology image. The density map images are shown in Figure 3.14 (c) and (d), from which it can be seen that there are more microglia cells particularly in the hippocampal region of the Alzheimer's Disease brain.



(a) (b)

(c) (d)

Figure 3.14: Analysis results of a typical (a) Alzheimer's mouse brain slice (b) healthy mouse brain slice. (c) and (d) Heat map of the microglia density image and the corresponding colour bar, representing the number of microglia within a square region.

3.2.5 Summary

In this chapter, automated image analysis methods were introduced for segmenting the microglia from histology images. Segmentation of both microglia process and soma was achieved through a variational method in combination with a fast split Bregman algorithm which overcomes the problems caused by inhomogeneity of histology images. Experiments showed that the proposed methods are accurate, thus eliminating the inter-rater variability seen with manual analysis, and scalable to analysing large microglial datasets. To the best of our knowledge, this is the first time that the Mumford-Shah total variation is used to extract microglia from histology images. For future work, the microglia segmentation framework described in this chapter will be tested on large histology datasets of healthy and diseased brains, along with ground truth images, to validate its sensitivity to disease progression and pro-inflammatory states and therefore a viable tool for studying microglial biology.

Chapter 4

Automated segmentation of OCT images

4.1 Introduction

Optical coherence tomography (OCT) is a powerful imaging modality that uses low coherence interferometry to provide high-resolution cross-sectional images of biological tissues, from which structural and molecular information of the tissues can be obtained [40]. Over the past two decades, OCT has become a well-established imaging modality and widely used by ophthalmologists for diagnosis of retinal and optical nerve diseases. One of the OCT imaging biomarkers for retinal and optical nerve disease is the thickness of the retinal layers. Automated OCT image segmentation to detect retinal layer boundaries is therefore required.

However, since the intensity patterns in OCT images are the result of light absorption and scattering in tissues, OCT images usually contain a significant amount of inhomogeneity and speckle noise, posing significant challenges to automated segmentation to identify tissue boundaries and other specific features. With retinal OCT imaging, disrupted retinal structures caused by pathologies and shadows by retinal blood vessels further complicate the segmentation process, leading to inaccuracy or failure of automated retinal layer segmentation algorithms.

In recent years many automatic and semi-automatic OCT segmentation approaches have been proposed. These approaches can be largely divided into three groups: A-scan based methods, B-scan based methods and volume based methods, as illustrated in Figure 4.1. A-scan based methods [133–136] detect intensity peak or valley points on the boundaries in each A-scan profile and then connect the detected points to form a continuous boundary using model fitting techniques. These methods can be inefficient

and inaccurate. B-scans methods [137–146, 131] outperform A-scan methods in general. However, they are prone to speckle noise in OCT images and likely to fail on pathological images. Common approaches to segmenting two-dimensional (2D) B-scans include active contour methods [147, 137–140], shortest-path based graph search [141–143] and statistical shape models [144–146] (i.e., active shape and appearance models [148, 149]). Popular volume based methods are graph based[150–156] and pattern recognition methods [157–160]. Computation of these methods can however be very complex and slow. Pattern recognition methods for retinal layer segmentation normally require manually segmented training data for classification. Automated segmentation of retinal layers from OCT images remains a challenge.

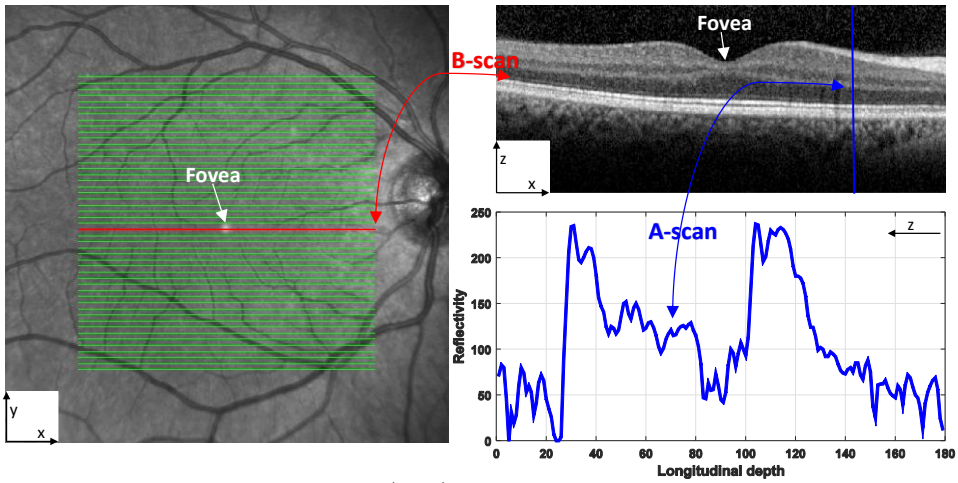


Figure 4.1: A en-face fundus image (left) overlaid with lines representing the locations of B-scans in a volumetric OCT image. The red line corresponds to the B-scan in the image (top right). One vertical A-scan of the B-scan is shown in the plot (bottom right). The fovea region is characterised by a depression in the centre of the retinal layer.

In this chapter, we present a new algorithm for retinal layer segmentation from OCT images based on a novel geodesic distance weighted by an exponential function. In contrast a single horizontal gradient used in other approaches [141, 155, 156], the exponential function in our method integrates both horizontal and vertical gradient information and can thus account for intensity variations in both directions. The exponential function also plays the role of enhancing weak retinal layer boundaries. As a result, the proposed geodesic distance method (GDM) is able to segment complex retinal structures with large curvatures and other irregularities caused by pathologies. We compute the weighted geodesic distance via an Eikonal equation using the fast sweeping method [10, 11, 161]. Retinal layer boundaries can then be detected using the geodesic distance by solving an ordinary differential equation via a time-dependent gradient descent. A local search region is identified based on the detected boundary to detect all the nine retinal layer boundaries

and overcome the local minima problem of the GDM. The retinal layer boundaries detected by the proposed GDM are shown in Figure 4.2. We evaluate the proposed GDM through extensive numerical experiments and compare it with state-of-the-art OCT segmentation approaches on both healthy and pathological images.

In the following sections, we will first review the state-of-the-art methods for comparison with the proposed GDM, such as parallel double snakes [140], Chiu's graph search [141], Dufour's method [153], and OCTRIMA3D [155, 156]. This will be followed by the details of the proposed GDM, ground-truth validation, numerical experiments, and comparison of the GDM with the above mentioned state-of-the-art methods.

4.2 Related work

In this section, we will limit our review of the state-of-the-art methods to only those that we will compare our GDM with in Section 4.3 (i.e., parallel double snakes [140], Chiu's method [141], OCTRIMA3D [155, 156], Dufour's method [153]). For a complete review on related subjects, we refer the reader to [162]. Among the four methods reviewed, the first two are for segmenting 2D B-scans, and the latter two are for segmenting 3D volumes.

Parallel double snakes (PDS): Rossant et al. [140] detected the pathological (retinitis pigmentosa) cellular boundaries in B-scan images by minimising an energy functional that includes two parallel active parametric contours. Their proposed PDS model consists of a centreline $C(s) = (x(s), y(s))$ parametrised by s and two parallel curves $C_1(s) = C(s) + b(s)n(s)$ and $C_2(s) = C(s) - b(s)n(s)$ with $b(s)$ being a spatially varying half-thickness and $n(s) = (n_x(s), n_y(s))$ the normal vector to the centreline $C(s)$. Specifically, their PDS model is defined as

$$E(C, C_1, C_2, b) = E_{Image}(C_1) + E_{Image}(C_2) + E_{Int}(C) + R\left(C_1, C_2, b\right), \tag{4.1}$$

where the image energy $E_{Image}(C_1) = -\int_0^1 |\nabla I(C_1)|^2 ds$ (∇ is the image gradient operator) attracts the parametric curve C_1 towards one of retinal borders of the input B-scan I, whilst $E_{Image}(C_2)$ handles curve C_2 which is parallel to C_1. The internal energy $E_{Int}(C) = \frac{\alpha}{2}\int_0^1 |C_s(s)|^2 ds + \frac{\beta}{2}\int_0^1 |C_{ss}(s)|^2 ds$ imposes both first and second order smooth regularities on the central curve C, with α and β respectively controlling the tension and rigidity of this curve. $R(C_1, C_2, b) = \frac{\varphi}{2}\int_0^1 |b'(C)|^2 ds$ is a parallelism constraint imposed on C_1 and C_2. Nine retinal borders have been detected by the method, i.e., ILM, RNFL_o, IPL-INL, INL-OPL, OPL-ONL, ONL-IS, IS-OS, OS-RPE and RPE-CH.

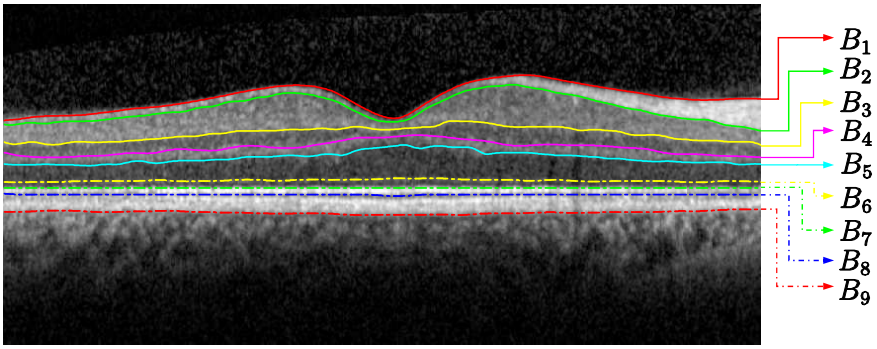


Figure 4.2: An example cross-sectional B-Scan OCT image centred at the macula, showing nine target intra-retinal layer boundaries detected by the proposed method. The names of these boundaries labelled as notations B_1,B_2...B_9 are summarised in Table 4.1.

Table 4.1: Notations for nine retinal boundaries/surfaces, their corresponding names and abbreviations

Notation	Name of retinal boundary/surface	Abbreviation
B_1	internal limiting membrane	ILM
B_2	outer boundary of the retinal nerve fibre layer	RNFL_o
B_3	inner plexiform layer-inner nuclear layer	IPL-INL
B_4	inner nuclear layer-outer plexiform layer	INL-OPL
B_5	outer plexiform layer-outer nuclear layer	OPL-ONL
B_6	outer nuclear layer-inner segments of photoreceptors	ONL-IS
B_7	inner segments of photoreceptors-outer segments of photoreceptors	IS-OS
B_8	outer segments of of photoreceptors-retinal pigment epithelium	OS-RPE
B_9	retinal pigment epithelium-choroid	RPE-CH

Chiu's method: Chiu et al. [141] considered retinal layer boundary detection in a B-scan image as determining the shortest-path that connects two points in a graph $G = (V, E)$, where V is the set of nodes in the graph corresponding to pixels in the B-scan image, and E is the set of weights assigned to pairs of nodes in the graph. Each node is connected only to its eight nearest neighbours, resulting in a sparse adjacency matrix of weights representing intensity variations in vertical direction. For example, an $M \times N$ sized image has an $MN \times MN$ sized adjacency matrix with $8MN$ non-zero entries. The weights are calculated from the intensity gradient of the image in vertical direction. Mathematically, the weights are calculated as

$$w(a,b) = \begin{cases} 2 - (g_a + g_b) + w_{\min} & \text{if } |a-b| \leq \sqrt{2} \\ 0 & \text{otherwise} \end{cases}, \tag{4.2}$$

where g is the vertical gradient of the B-scan image; a and b denote two separate nodes in V respectively and w_{min} is a small positive value to stabilise the system. The most

prominent layer boundary is then detected as the minimal weighed path from the first to the last vertex in V using the Dijkstra's algorithm. A similar region refinement technique to Section 4.3.4 was used to detect seven retinal boundaries, i.e., ILM, RNFL$_o$, IPL-INL, INL-OPL, OPL-ONL, IS-OS and RPE-CH.

Dufour's method: Dufour et al. [153] proposed a modification of optimal graph search approach [163] to segment retinal layers in 3D OCT images. By using soft constraints and prior knowledge, they improve the accuracy and robustness of the original framework. Specifically, their Markov random field based model is given by

$$E(S) = \sum_{i=1}^{n} \left(E_{boundary}(S_i) + E_{smooth}(S_i)\right) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E_{inter}(S_i, S_j),$$

where S is a set of layer boundaries S_1 to S_n. The external surface energy $E_{boundary}(S_i)$ is computed from 3D OCT images. The surface smoothness energy $E_{smooth}(S_i)$ guarantees the connectivity and regularises the layers. The interaction energy $E_{\text{inter}}(S_i, S_j)$ integrates soft constraints to regularise the distances between two simultaneously segmented layer boundaries. This model is then built from training datasets consisting of fovea-centered OCT slice stacks. Their method is capable to segment six retinal layers ($n = 6$ in above formulation) in both healthy and macular edema subjects, i.e., ILM, RNFL$_o$, IPL-INL, OPL-ONL, IS-OS and RPE-CH.

OCTRIMA3D: Tian et al. [155, 156] proposed a real-time automatic segmentation method for 3D OCT images. The segmentation was done frame-by-frame in each 2D B-Scan by considering the spatial dependency between each two adjacent frames. Their work is based on Chiu's graph search framework [141] for B-Scan OCT images, with added inter-frame flattening to reduce the curvature in the fovea region and thus improving the accuracy. They also use inter-frame or intra-frame information to limit the search region in current or adjacent frame so as to increase the computational speed. The method can segment eight retinal layers, i.e., ILM, RNFL$_o$, IPL-INL, INL-OPL, OPL-ONL, IS-OS, OS-RPE and RPE-CH. Table 4.2 reports the retinal boundaries detected by the four methods as well as our GDM method detailed in the next section.

Table 4.2: Target boundaries of the five methods compared in this chapter (check mark means the layer boundary can be detected, while cross mark means the boundary cannot be detected).

Method	ILM (B_1)	RNFL$_o$ (B_2)	IPL-INL (B_3)	INL-OPL (B_4)	OPL-ONL (B_5)	ONL-IS (B_6)	IS-OS (B_7)	OS-RPE (B_8)	RPE-CH (B_9)
PDS [140]	✓	✓	✓	✓	✓	✓	✓	✓	✓
Chiu's method [141]	✓	✓	✓	✓	✓	×	✓	×	✓
Dufour's method [153]	✓	✓	✓	×	✓	×	✓	×	✓
OCTRIMA3D [155, 156]	✓	✓	✓	✓	✓	×	✓	✓	✓
Proposed GDM	✓	✓	✓	✓	✓	✓	✓	✓	✓

4.3 Proposed geodesic distance method

In this section, we present our GDM method for segmentation of OCT images to detect nine retinal layers defined in Figure 4.2 and Table 4.1. We will describe the method for 2D segmentation in detail. For 3D segmentation, we first calculate the 3D geodesic distance volume in a manner similar to that of 2D segmentation, followed by minimal path detection on each slice of the geodesic distance volume. In Appendix B, we present the implementation details of geodesic distance calculation for both 2D and 3D segmentation.

4.3.1 Geodesic distance

We use geodesic distance to identify the pixels on the boundaries of retinal layers in OCT images. The geodesic distance d is the smallest integral of a weight function W over all possible paths from two points s_1 and s_2. The weight function determines how the path goes from s_1 to s_2. Small weight at one point indicates that the path has high possibility of passing that point. Specifically, the weighted geodesic distance between two endpoints s_1 and s_2 is given by

$$D\left(s_1, s_2\right) = \min_{C} \int_0^1 W^{-1}\left(C\left(s\right)\right) ds. \tag{4.3}$$

Above $C\left(s\right)$ is the set of all possible paths that link s_1 to s_2, the path length is normalised to unity, and the start and end locations are $C(0) = s_1$ and $C(1) = s_2$, respectively. The infinitesimal contour length ds is weighted by a non-negative function $W\left(C\left(s\right)\right)$. This minimisation problem can be interpreted as finding a geodesic curve (i.e., a path with the smallest weighted length) in a Riemannian space. It is known that the solution of (4.3) satisfies the Eikonal equation (4.5).

The retinal layer boundaries in OCT images are normally near horizontal so the largest intensity changes at the layer boundaries are likely to be in the vertical direction. The gradient in the vertical direction is thus taken to compute weight W in (4.3) in order to determine the path that passes the points with maximum gradient changes. For instance, each of the two prominent boundaries, e.g., ILM (B_1) and IS-OS (B_7) in Figure 4.3 (a) and (e), is at the border of a dark layer above a bright layer. As a result, pixels in the region around the two boundaries have high gradient values, as shown in Figure 4.3 (b) and (f). As the retinal layers at each side of the boundary are either transiting from dark to bright or bright to dark, the non-negative weight function W in this chapter is defined based on intensity variation as follows

$$W\left(x\right) = \begin{cases} 1 - \mathbf{exp}\left(-\lambda\left(1 - n\left(\nabla_x I\right)\right) n\left(\left|\nabla_y I\right|\right)\right) & \text{dark-to-bright} \\ \mathbf{exp}\left(-\lambda\left(1 - n\left(\nabla_x I\right)\right) n\left(\left|\nabla_y I\right|\right)\right) & \text{bright-to-dark} \end{cases}, \tag{4.4}$$

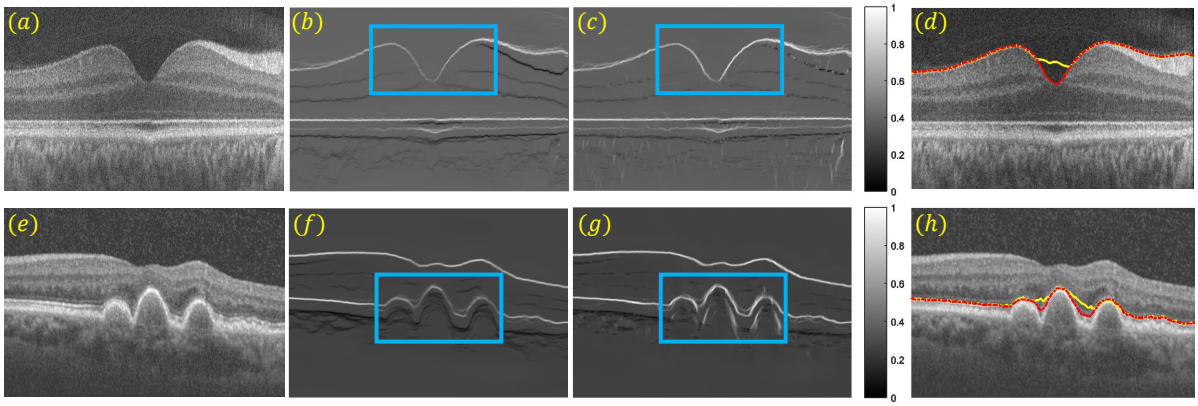


Figure 4.3: Effectiveness of the weight W defined in (4.4) on real OCT images. (a) and (e): normal B-scan OCT data and pathological B-scan from an eye with age-related macular degeneration (drye-AMD); (b) and (f): vertical dark-to-bright gradient maps of (a) and (e), respectively; (c) and (g): dark-to-bright gradient maps calculated using equation (4.4) with $\lambda = 1$. Note that the pixel gradients have been enhanced in the blue rectangular region where large curvature and bumps occur; (d) and (h): layer boundary detection results using the method described in Section 4.3.3 with different gradient maps: the yellow lines are computed using (b) and (f), and the red lines using (c) and (g).

where I is an input OCT image; $n(\cdot)$ is a linear stretch operator used to normalise values to between 0 and 1; **exp** is the exponential function, and λ is a user-define parameter, which together enhance the foveal depression regions and highlight the weak boundaries [164]; ∇_x and ∇_y are the first-order gradient operator along x (vertical) and y (horizontal) direction respectively, which are discretised using a central finite difference scheme under the Neumann boundary condition; and $n(|\nabla_y I|)$ is positive horizontal gradient, without which only vertical direction is accounted for and is thus only applicable to flat boundaries, as evident in Figure 4.4. Consequently, the GDM with the weight W defined in (4.4) is robust against curved features (e.g., the central region of the fovea) as well as other irregularities (e.g., bumps or large variations of boundary locations) caused by pathologies, as illustrated in Figure 4.3 as well as in the experimental section.

4.3.2 Selection of endpoints s_1 and s_2

For fully automated segmentation, it is essential to find a way to initialise the two endpoints s_1 and s_2 automatically. Since the retinal boundaries in the OCT images used in this chapter run across the entire width of the image, we add an additional column on each side to the gradient map computed from (4.4). As the minimal weighted path is sought after, a weight W_{max} larger than any of the non-negative weights calculated

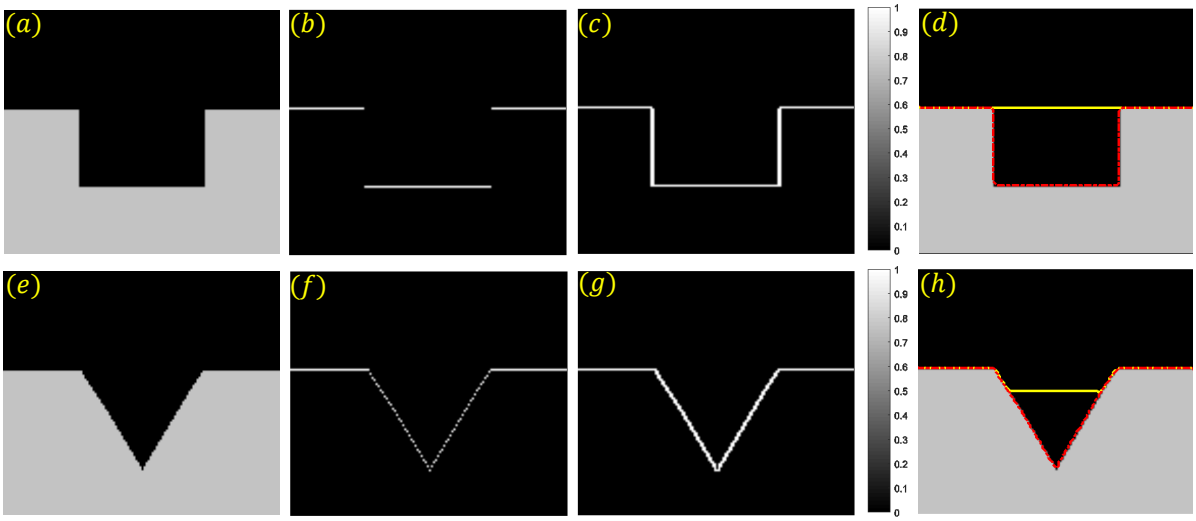


Figure 4.4: Eeffectiveness of the weight W defined in (4.4) on two synthetic images. (a) and (e): synthetic images with changes in both vertical and horizontal directions; (b) and (f): pure vertical dark-to-bright gradient maps of (a) and (e), respectively; (c) and (g): dark-to-bright gradient maps calculated using equation (4.4) with $\lambda = 1$ - both vertical and horizontal gradients are enhanced using the proposed method, leading to robust gradient maps for segmentation; (d) and (h): boundary detection results via the method described in Section 4.3.3 using different gradient weights. Yellow lines are computed using (b) and (f), whilst red lines using (c) and (g).

from (4.4) is therefore assigned to each of the newly added vertical columns (note that we use W^{-1} for the geodesic distance (4.3), the minimal weighted path thereby prefers large weights). This forces the path traversal in the same direction as the newly added vertical columns with maximal weights, and also allows the start and end points to be arbitrarily assigned in the two columns. Once the retinal layer boundary is detected, the two additional columns can be removed. Row 1 and 2 in Figure 4.5 respectively show different initialisations of two endpoints as well as the corresponding path evolution results.

4.3.3 Eikonal equation and minimal weighted path

The solution of (4.3) can be obtained by solving the Eikonal equation after the endpoints are determined. Specifically, over a continuous domain, the distance map $D(x)$ to the seed point s_2 is the unique solution of the following Eikonal equation in the viscosity sense

$$|\nabla D(x)| = W^{-1}(x), \; x \in \Omega \backslash \{s_2\} \tag{4.5}$$

and

$$D(x) = 0,\ x \in \{s_2\}.$$

The equation is a first order nonlinear partial differential equation and its solution can be found via the classical fast marching algorithm [8, 9] using an upwind finite difference approximation with the computational complexity $O(MNlog(MN))$ (MN is the total number of grid points). Recently, the fast sweeping algorithm [10, 11] has been proposed. This technique is based on a pre-defined sweep strategy, replacing the heap priority queue to find the next point to process, and thereby has the linear complexity of $O(MN)$. Fast sweeping is faster than fast marching for simple geometry problems. However, the situation may be reversed for complex geometry. In this chapter, we apply fast sweeping for (4.5) and its detailed implementation has been given in Appendix B. Figure 4.6 shows two distance maps calculated using the weight and end points defined in Figure 4.5.

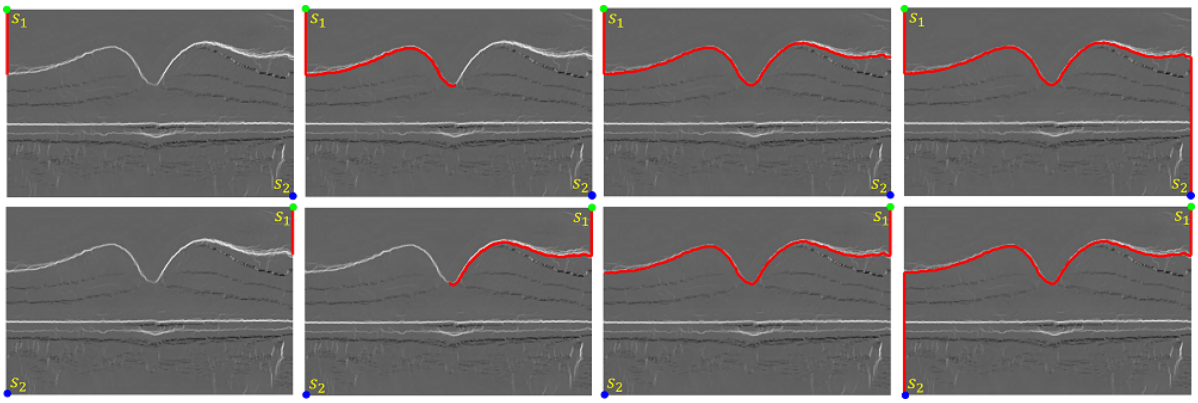


Figure 4.5: Two set of segmentation examples using different automatic endpoints initialisations on a dark-to-bright gradient map. s_1 and s_2 are start and end points, respectively. Row 1 and 2 respectively show the path evolution results using Equation (4.7). The paths start at s_1 and end at s_2.

Once the geodesic distance map to the end point s_2 is found, the minimal weighted path (geodesic curve γ) between point s_1 and s_2 can be extracted from the following ordinary differential equation through the time-dependent gradient descent

$$\gamma'(t) = -\eta_t \nabla D(\gamma(t)), \quad \gamma(0) = s_1, \tag{4.6}$$

where $\eta_t > 0$ controls the parametrisation speed of the resulting curve. To obtain unit speed parametrisation, we use $\eta_t = |\nabla D(\gamma(t))|_{\varepsilon}^{-1}$. Since the distance map D is nonsmooth at point s_2, a small positive constant ε is added to avoid dividing by zero. Note that γ is guaranteed to end at the point s_2 by solving the ordinary differential equation, because the distance field is monotonically decreasing from s_1 to s_2, as observed

in Figure 4.6. This technique can achieve sub-pixel accuracy for the geodesic path even if the grid is discrete.

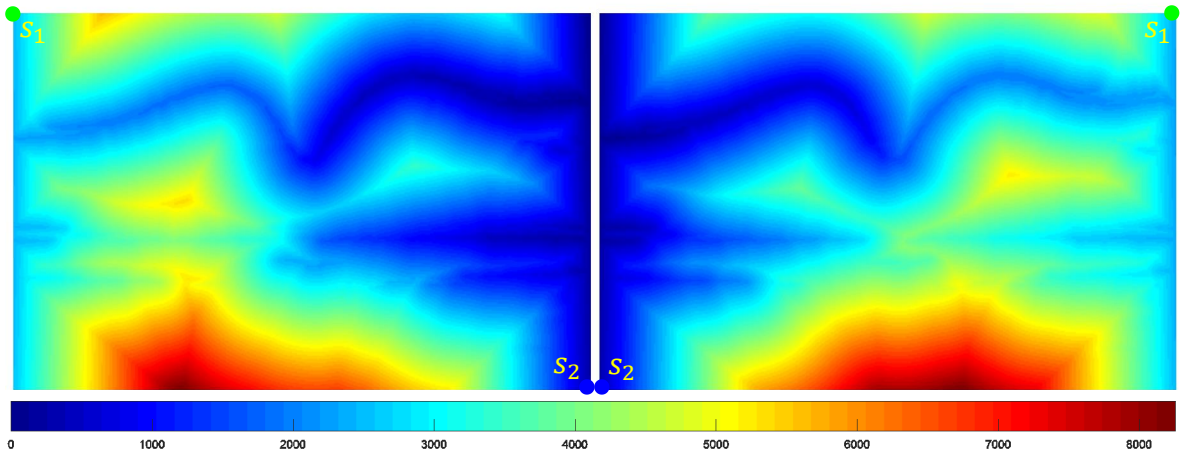


Figure 4.6: Distance maps calculated using the fast sweeping method on the gradient weights in Figure 4.5. The left distance map is computed using the end point s_2 in the 1st row of Figure 4.5, while the right distance map using the end point s_2 in the 2nd row of Figure 4.5. The range of distance values is represented by the color bar at the bottom.

The geodesic curve is then numerically computed using a discretised gradient descent, which defines a discrete curve γ^k using

$$\gamma^{k+1} = \gamma^k - \tau G\left(\gamma^k\right), \tag{4.7}$$

where γ^k is a discrete approximation of $\gamma(t)$ at time $t = k\tau$, and the time step size $\tau > 0$ should be small enough. $G(x)$ is the normalised gradient $\nabla D(\gamma(t))/|\nabla D(\gamma(t))|_\varepsilon$ parametrised by the arc length. Once γ^{k+1} reaches s_2, one of the retinal boundaries can be found. The following **Algorithm 1** concludes the proposed GDM for extracting one retinal border from an OCT B-scan.

Algorithm 1: Proposed GDM for one retinal boundary detection

1: Input OCT B-scan data I
2: calculate dark-to-bright or bright-to-dark weight W using (4.4)
3: pad two new columns to the weight and assign large values to them
4: select two endpoints s_1 and s_2 on the two newly padded columns
5: calculate distance map D in (4.5) using fast sweeping algorithm
6: find one retinal layer boundary γ using the gradient descent flow (4.7)
7: remove the additional columns in the edge detection result

4.3.4 Detection of nine retinal layer boundaries

In this section, we show the implementation details of the proposed approach to segment nine retinal layer boundaries, as shown in Figure 4.2 and Table 4.1. Since the proposed model (4.3) is not convex due to the image gradient used, its solution can easily get stuck in local optima. For example, Figure 4.3 (c) and (g) have high gradient values in the region around both the ILM and IS-OS boundaries. However, in Figure 4.3 (d) the algorithm detected the ILM boundary while in Figure 4.3 (h) it detected IS-OS. In order to eliminate such uncertainty, we dynamically define the search region based on the detected boundaries. The following describes the detection of the nine boundaries in a hierarchical fashion.

Detection of the IS-OS boundary

The intensity variation between two layers divided by the IS-OS (B_7) border are normally the most prominent in OCT B-scans. However, since OCT images are always corrupted by speckle noise as a result of light absorption and scattering in the retinal tissue, it is not always the case. For example, the intensity variation around the IML (B_1) border sometimes can be more obvious than that around IS-OS, as shown in the gradient image Figure 4.3 (c). To make sure the first segmentation being the IS-OS boundary we first enhance it via a simple local adaptive thresholding approach[1], which is given as follows

$$p = \begin{cases} 0 & ls\left(I, ws\right) - I > C \\ 1 & \text{otherwise} \end{cases}, \tag{4.8}$$

where I is the input OCT image, and $ls\left(p, ws\right)$ means that I is convolved with a suitable operator, i.e., the mean, Gaussian or median filter. We mention that the purpose of using a filter in (4.8) is to overcome the inhomogeneity effect by oversmoothing the input image such that the resulting image can be simply segmented by the thresholding method. Our previous research [92, 94] has shown that such a method is very robust against intensity inhomogeneity appeared in different medical images. ws is the window size of the filter and C a user-defined threshold value. In the chapter, we use the mean filter with window size $ws = 100$ and set $C = 0.01$. The enhanced image can then be obtained by multiplying the original image I with p. The first two images in Figure 4.7 illustrate that the contrast of the IS-OS border has been enhanced and the most obvious intensity variation now takes place around the IS-OS layer boundary. The IS-OS boundary is

[1] http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm

then detected on a dark-to-bright gradient image. Consequently, the detected line is guaranteed to pass IS-OS, as shown in the last two images in Figure 4.7.

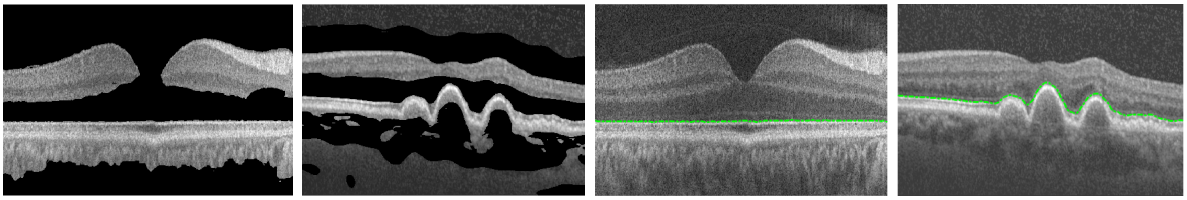


Figure 4.7: Detecting the IS-OS boundaries in the normal and pathological images after image enhancement via a local adaptive thresholding method (4.8).

Detection of the RPE-CH, OS-RPE and ONL-IS boundaries

Once IS-OS (B_7) is segmented, it can be used as a reference to limit the search region for segmenting the RPE-CH (B_9), OS-RPE (B_8) and ONL-IS (B_6) boundaries. RPE-CH and OS-RPE are below IS-OS and they are detected in the following way: RPE-CH can be extracted by applying the GDM on the bright-to-dark gradient weight obtained from the region pixels below IS-OS (i.e., the bright-to-dark weight is set to zeros above IS-OS); OS-RPE is then detected on the bright-to-dark gradient weight in the region between the IS-OS and RPE-CH boundaries (i.e., the bright-to-dark gradient weight is set to zeros outside of the region between IS-OS and RPE-CH). ONL-IS is above IS-OS. The search region can be constructed between IS-OS and a parallel line above it with a diameter of 15 pixels. The dark-to-bright gradient weight outside of the region is then set to zeros. Hence, the only layer boundary in the search region is ONL-IS which can be extracted using the GDM on the dark-to-bright gradient weight.

Detection of the ILM and INL-OPL boundaries

Both ILM (B_1) and INL-OPL (B_4) are at the border of a darker layer above a bright layer. The intensity variation around the IML boundary is much more prominent and thus it is segmented first. The detected ONL-IS boundary is taken as a reference and the dark-to-bright gradient weight below ONL-IS is set to zeros. INL-OPL can be then easily detected on the dark-to-bright gradient weight by simply limiting the search region between ILM and ONL-IS (i.e., the dark-to-bright gradient weight is set to zeros outside of the region between ILM and ONL-IS).

Detection of the OPL-ONL, IPL-INL and RNFL$_o$ boundaries

OPL-ONL (B_5), IPL-INL (B_3) and RNFL$_o$ (B_2) demonstrate a bright layer above a darker layer and thus can be detected on the bright-to-dark gradient weight. The segmented INL-OPL and ONL-IS boundaries are taken as two reference boundaries, and OPL-ONL can be found by limiting the search region between INL-OPL and ONL-IS. The search region for IPL-INL can be then constructed between the INL-OPL boundary and a parallel line above it with a diameter of 20 pixels. IPL-INL can be located on a bright-to-dark gradient weight which is set to zeros outside of the search region constructed. Finally, RNFL$_o$ can be found in the search region between the two reference boundaries IPL-INL and IML. However, because the IPL-INL and IML boundaries are very close to each other in the central region of the fovea, the search region for the RNFL$_o$ are sometimes missing around the fovea region. This leads to segmentation errors of RNFL$_o$, as shown in Figure 4.8 (a). These errors however can be avoided by simply removing the spurious points detected on RNFL$_o$ in the region above IML, as shown in Figure 4.8 (b). For clarity, the proposed method for segmenting nine retinal layer boundaries is summarised in the flow chart shown in Figure 4.9.

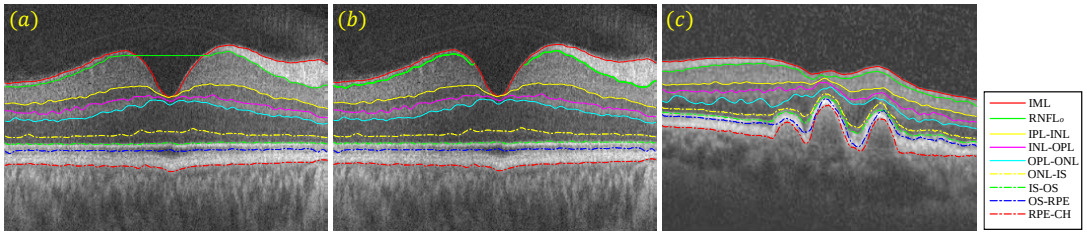


Figure 4.8: Segmentation results of the nine retinal layer boundaries on both normal and dye-AMD pathological B-scans, as shown in (a) and (c). The detection of the RNFL$_o$ boundary however shows errors due to the absence of a search region for this boundary in, as evident in (a). (b) shows that these errors have been corrected.

4.4 Experiment setup

To evaluate the performance of the proposed GDM qualitatively and quantitatively, numerical experiments are conducted to compare it with the state-of-the-art approaches reviewed in Section 4.2 on both healthy and pathological OCT retinal images. As the GDM is able to segment both 2D and 3D OCT images, we perform numerical experiments on both B-scans and volumetric OCT images. A pre-processing method [68] is used to reduce noise prior to determining the layers boundaries for all segmentation methods. In the following, we introduce the detailed procedure of OCT data acquisition, evaluation

metrics used to quantify segmentation results, final numerical results, and computational complexity of different methods.

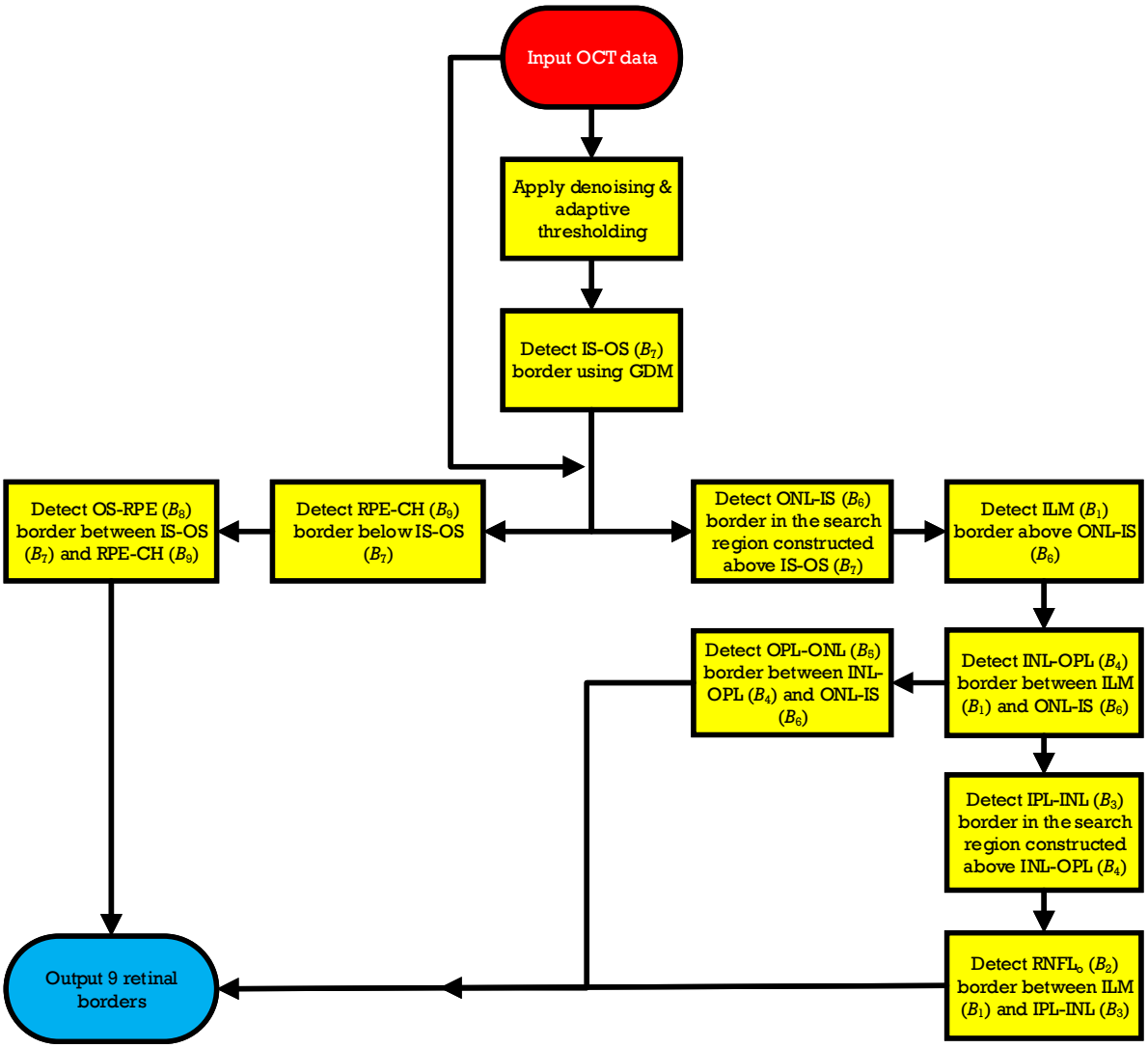


Figure 4.9: The overview of the proposed framework for dynamically detecting nine retinal layer boundaries defined in Figure 4.2 and Table 4.1. Section 4.3.4 describes this flow chart in detail.

4.4.1 Clinical data

30 Spectralis SDOCT (ENVISU C class 2300, Bioptigen, axial resolution = 3.3µm, scan depth = 3.4mm, 32, 000 A-scans per second) B-scans from 15 healthy adults (mean age = 39.8 years, SD = 8.6 years; 7 male, 8 female) were used for the research. All the data

was collected after informed consent was obtained and the study adhered to the tenets of the Declaration of Helsinki and Ethics Committee approval was granted.

2D B-scan data: The B-scan was imaged from the left and right eye of 15 healthy adults using a spectral domain OCT device with a chin rest to stabilise the head. The B-scan located at the foveal centre was identified from the lowest point in the foveal pit where the cone outer segments were elongated (indicating cone specialisation). To reduce the speckle noise and enhance the image contrast, every B-scan was the average of aligned images scanned at the same position. In addition to the 30 OCT images from the healthy subjects, another 20 B-scans from subjects with pathologies are also used to compare the proposed GDM with other approaches in pathological cases. These B-scans are from an eye with dry age-related macular degeneration (drye-AMD), which is available from the Dufour's software package's website[2]. Segmentation accuracy by the three automated 2D methods (i.e., PDS, Chiu's method and GDM) over these healthy and pathological B-scans is evaluated using the ground truth datasets, which were manually detected with carefulness by one observer.

3D OCT data: 10 Spectralis SD-OCT (Heidelberg Engineering GmbH, Heidelberg, Germany) volume data sets from 10 healthy adult subjects are used in this study. Each volume contains 10 B-scans, and the OCT A-scans outside the 6mm × 6mm (lateral × azimuth) area and centred at the fovea were cropped to remove low signal regions. All volumetric data can be downloaded from [155], where also contains the results of the **OCTRMA3D**, and the manual labellings from two graders. In this study we choose the manual labelling of grader 1 as the 3D ground truth.

4.4.2 Evaluation metrics

Performance metrics are defined to demonstrate the effectiveness of the proposed GDM and compare it with the existing methods. Three commonly used measures of success for retinal layer boundary detection are signed error (SE), absolute error (AE) and Hausdorff distance (HD). Among them, SE indicates the bias and variability of the results. AE is the absolute difference between automatic segmentation and ground truth, while HD measures the distance between the farthest point of a set to the nearest point of the

[2]http://pascaldufour.net/Research/software_data.html

other and vice versa. Specifically, these metrics are denoted as

$$\begin{aligned}
\mathrm{SE}\left(B_i, \tilde{B}_i\right) &= \tfrac{1}{n}\sum_{j=1}^{n}\left(B_{ij} - \tilde{B}_{ij}\right), \\
\mathrm{AE}\left(B_i, \tilde{B}_i\right) &= \tfrac{1}{n}\sum_{j=1}^{n}\left(\left|B_{ij} - \tilde{B}_{ij}\right|\right), \\
\mathrm{HD}\left(B_i, \tilde{B}_i\right) &= \max\left(\max_{x\in B_i}\left\{\min_{y\in\tilde{B}_i}\|x-y\|\right\}, \max_{x\in\tilde{B}_i}\left\{\min_{y\in B_i}\|x-y\|\right\}\right).
\end{aligned}$$

Above B_i and $\tilde{B}_i$ are respectively the detected boundaries and ground truth boundaries (i.e., manual labellings). n is the number of pixels/volexs that fall on the retinal layer boundary. In our comparison, when the SE value is close to zero, the difference between B_i and $\tilde{B}_i$ is normally small. In this case, the result is less biased. The measurements of AE and HD (varies from 0 to ∞ theoretically) signify the difference between two boundaries, e.g., 0 indicates that both retinal structures share exactly the same boundary, and larger AE and HD values mean larger distances between the measured boundaries. We also monitor the overall SE (OSE), AE (OAE) and HD (OHD) during all the experiments. They are defined as

$$\begin{aligned}
\mathrm{OSE} &= \frac{1}{s}\sum_{i=1}^{s}\mathrm{SE}\left(B_i, \tilde{B}_i\right), \\
\mathrm{OAE} &= \frac{1}{s}\sum_{i=1}^{s}\mathrm{AE}\left(B_i, \tilde{B}_i\right), \\
\mathrm{OHD} &= \frac{1}{s}\sum_{i=1}^{s}\mathrm{HD}\left(B_i, \tilde{B}_i\right).
\end{aligned}$$

Here s is the total number of retina boundaries one method can detecte.

4.4.3 Parameter selection

There are five parameters in the PDS model: three smooth parameters α, β, φ and two time step sizes γ_C and γ_b used within the gradient descent equations to minimise the functional (4.1) with respect to C and b. In this chapter we use $\alpha = 10$, $\beta = 0$, $\varphi = 700$, $\gamma_C = 10$ and $\gamma_b \geq 2$ suggested in [140]. In addition, as PDS is a nonconvex model and its segmentation results depend on initialisation. We initialise the parallel curves very closely to the true retinal boundaries for fair comparison with other methods. A maximal number of iterations number 500 is used to ensure convergence of the PDS model. The graph theoretic based methods, i.e., Chiu's method, OCTRIMA3D and Dufour's method,

require no parameter input. Finally, our GDM has two build-in parameters: λ in (4.4) and τ in (4.7). We set $\lambda = 10$ and $\tau = 0.8$ to detect the retinal layers in the OCT images.

4.4.4 Numerical results

We first visually compare the segmentation results of the GDM, PDS and Chiu's graph search method on both healthy and pathological B-scans, which are shown in Figure 4.10 (a)-(d). The PDS results shown in (e)-(h) have some errors on some of detected boundaries. For instance, the detected B_1 and B_2 boundaries cannot converge to the true retinal boundaries around the central fovea region, as shown in (f) and (h). This is because PDS is the classical nonconvex snake-driven model which has difficulty handling concave boundaries. Moreover, because the B_7 retinal layer has a much stronger image gradient than the B_6 and B_8 layers, some parts of the segmented B_6 and B_8 boundaries have been mistakenly attracted to the B_7 layer. Since Chiu's graph search method merely considers intensity changes in the vertical direction (4.2), it also fails to segment the fovea region layers with strong curvature, as shown in (j) and (l). Moreover, the algorithm cannot handle the irregular bumps caused by pathologies very well, as observed from the bottom B_9 boundaries detected in (k) and (l). In general, Chiu's method works very nicely when retinal structures are flat or smooth without large changes at boundary locations. As compared to the ground truth in the last row, the results by the proposed GDM method are the best, as shown in (m)-(p). As analysed in Section 4.3, the gradient weights defined in (4.4) account for both vertical and horizontal variations, making it very suitable for both flat and nonflat retinal structures. Hence, GDM is a better clinical tool for detecting retinal layer boundaries from normal and pathological images.

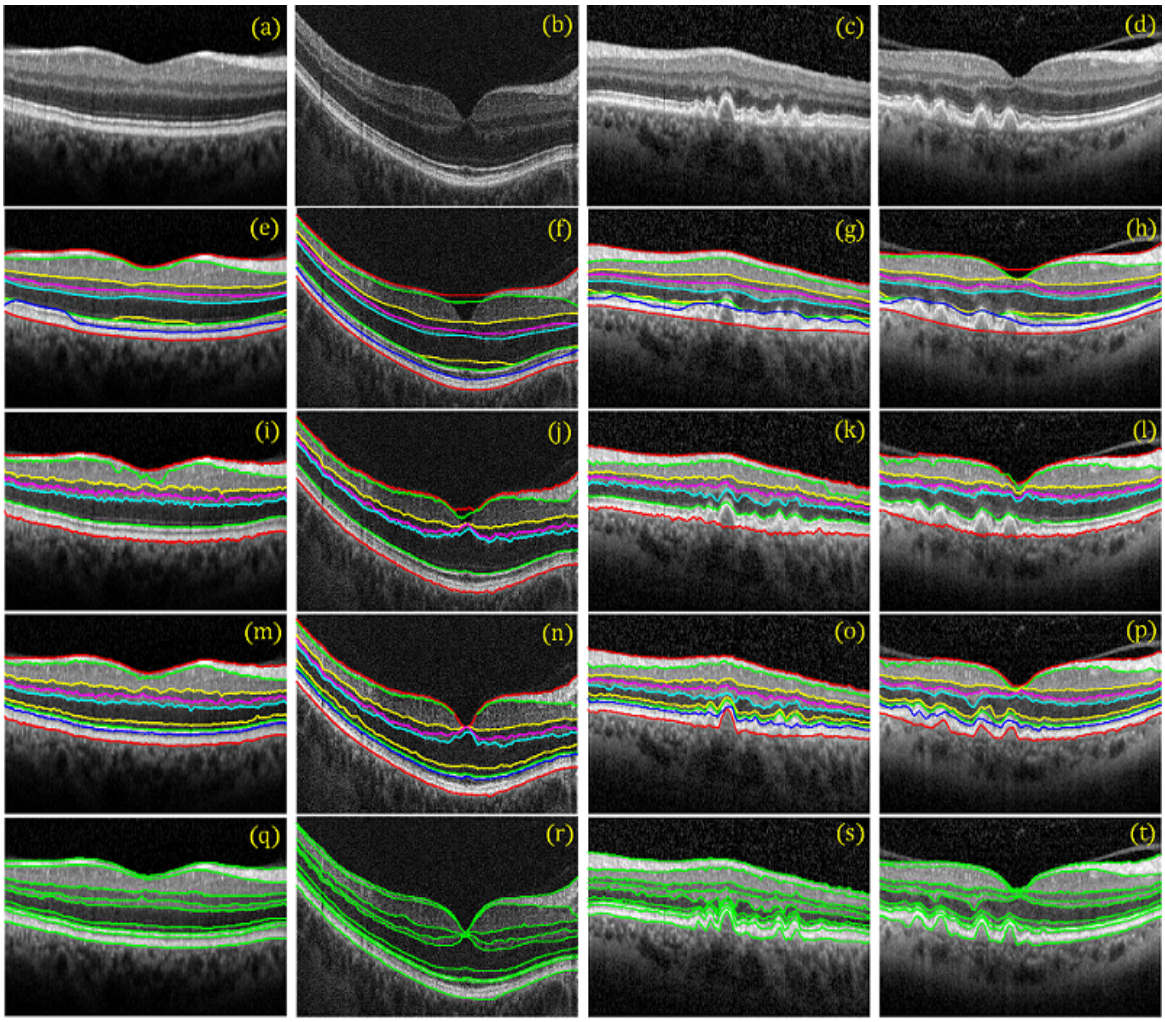


Figure 4.10: Comparison of different OCT segmentation methods using healthy and pathological B-scans. 1st row: original healthy B-scans (first two columns) and pathological B-scans (last two columns); 2nd row: results by the PDS model (4.1); 3rd row: results by Chiu's method; 4th row: results by the proposed GDM; 5th row: ground truth.

The accuracy of the segmentation results by different methods against ground truth on 30 healthy and 20 pathological B-scans is indicated in Table 4.3 and Table 4.4, respectively. In order to make the comparison clearer, we plot the data in the tables in Figure 4.11 and Figure 4.12, respectively.

Table 4.3: Mean and standard deviation of SE (μm), AE (μm) and HD (μm) calculated using the results of different methods (PDS, Chiu' method and GDM) and the ground truth manual segmentation, over 30 healthy OCT B-scans. • (○) indicates that GDM is better (worse) than the compared methods (paired t-tests at 95% significance level).

	SE (μm)			AE (μm)			HD (μm)		
Boundary	PDS	Chiu et al.	GDM	PDS	Chiu et al.	GDM	PDS	Chiu et al.	GDM
ILM (B_1)	-3.92±1.90•	-1.22±0.68•	0.273±0.33	4.615±2.03•	2.605±1.12•	0.924±0.26	36.56±15.9•	22.12±9.23•	3.702±1.62
RNFL$_o$ (B_2)	-2.57±1.38•	-1.67±1.34•	-0.53±0.37	3.864±1.49•	2.676±0.82•	1.262±0.34	29.00±11.6•	21.25±5.98•	7.340±2.16
IPL-INL (B_3)	-0.55±0.83○	-1.04±1.21•	-0.38±0.61	1.876±0.60•	2.020±0.79•	1.314±0.32	8.619±3.77•	10.53±5.25•	7.258±1.92
INL-OPL (B_4)	0.012±0.58•	-0.90±0.61○	-0.71±0.71	1.708±0.39○	1.699±0.40○	1.807±0.51	6.772±2.53○	7.036±2.84○	7.505±2.96
OPL-ONL (B_5)	-0.23±1.29•	-1.51±1.30•	-1.12±1.17	2.127±1.00•	2.133±1.05•	1.949±0.94	10.22±3.70 •	9.044±3.48•	7.463±3.24
ONL-IS (B_6)	6.010±0.83•	—	-0.73±0.49	6.055±0.86•	—	1.376±0.36	9.969±1.58•	—	4.630±1.05
IS-OS (B_7)	-0.09±0.61•	0.194±0.49○	0.291±0.63	0.823±0.29•	0.720±0.25○	0.771±0.36	3.676±1.63•	3.240±1.60•	2.611±0.74
OS-RPE (B_8)	5.202±2.25•	—	-0.78±0.47	5.570±1.76•	—	1.125±0.36	8.913±2.28•	—	3.601±0.96
RPE-CH (B_9)	-0.31±0.79•	-0.84±0.58•	-0.74±0.69	1.291±0.25•	1.228±0.47•	1.213±0.45	4.237±1.47•	4.027±1.31•	3.831±1.08
Overall	0.394±0.39•	-1.00±0.54•	-0.49±0.23	3.103±0.74•	1.869±0.59•	1.305±0.32	13.11±4.25•	11.04±3.75•	5.327±1.11

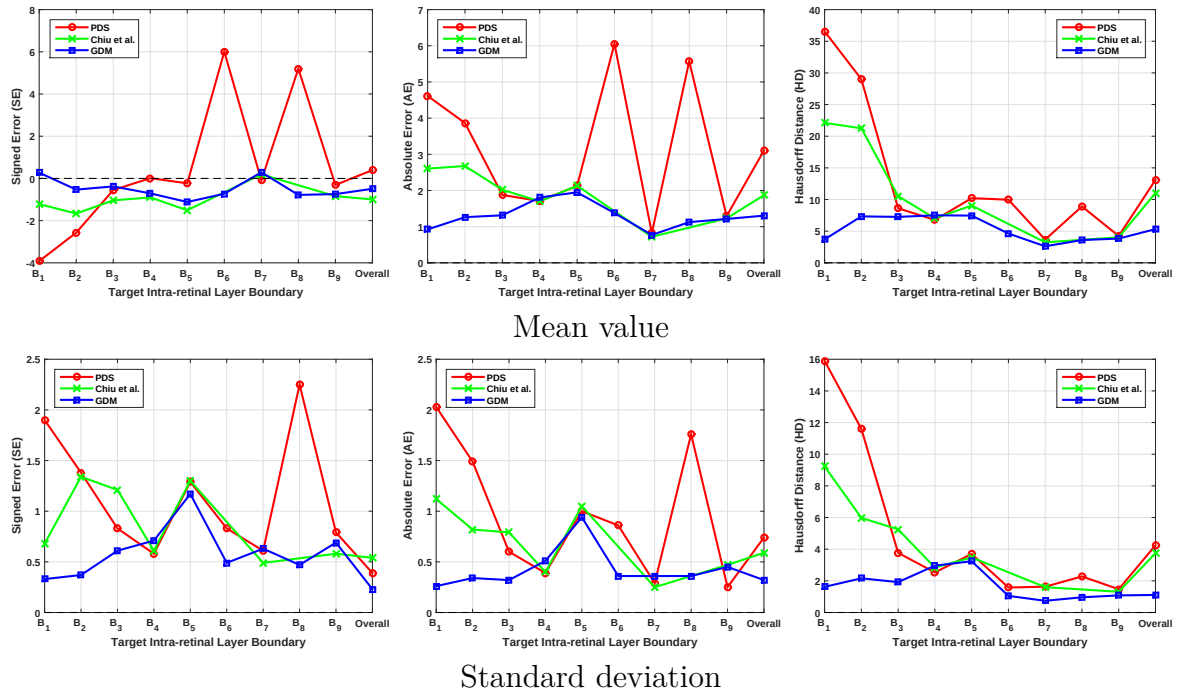


Figure 4.11: Plots of mean and standard derivation obtained by different methods in Table 4.3 for healthy B-scans. The 1st and 2nd rows respectively show the mean and standard derivation of SE (μm), AE (μm) and HD (μm) for segmenting boundaries $B_1 - B_9$ using PDS, Chiu's method and GDM. The overall value is the average result over all boundaries.

Table 4.4: Mean and standard deviation of SE (μm), AE (μm) and HD (μm) calculated using the results of different methods (PDS, Chiu's method and GDM) and the ground truth manual segmentation, over 20 pathological OCT B-scans. • (○) indicates that GDM is better (worse) than the compared methods (paired t-tests at 95% significance level).

	SE (μm)			AE (μm)			HD (μm)		
Boundary	PDS	Chiu et al.	GDM	PDS	Chiu et al.	GDM	PDS	Chiu et al.	GDM
ILM (B_1)	-0.41±0.59•	-0.34±0.25○	-0.36±0.29	0.932±0.44•	0.796±0.17•	0.683±0.09	6.461±4.86•	4.087±1.01•	3.337±1.10
RNFL$_o$ (B_2)	-0.93±0.93•	-0.38±0.33•	-0.49±0.50	1.792±0.63•	1.717±0.53•	1.257±0.32	6.145±1.84•	8.464±4.55•	6.109±2.49
IPL-INL (B_3)	-0.23±0.62○	-0.22±0.27○	-0.32±0.32	1.228±0.21•	1.149±0.20•	0.926±0.16	7.640±1.31•	5.857±0.98•	5.151±1.82
INL-OPL (B_4)	0.578±0.64•	0.555±0.39•	0.392±0.26	1.546±0.28•	1.563±0.30•	1.419±0.16	7.165±1.07•	8.194±1.36•	5.942±1.32
OPL-ONL (B_5)	-0.04±1.08○	0.286±0.55•	-0.07±0.64	2.371±0.76•	2.255±0.60•	2.019±0.65	11.28±1.95•	9.858±2.76•	9.281±2.25
ONL-IS (B_6)	3.339±1.22•	—	-0.57±0.72	4.484±0.50•	—	1.442±0.34	15.23±4.03•	—	6.205±1.01
IS-OS (B_7)	-0.23±0.86•	1.030±1.06•	0.350±0.50	2.415±1.25•	2.399±1.05•	1.055±0.22	15.95±10.2•	17.66±11.3•	6.795±4.65
OS-RPE (B_8)	2.371±4.17•	—	0.028±0.41	5.927±2.34•	—	1.821±0.47	22.63±12.9•	—	9.673±1.30
RPE-CH (B_9)	3.315±2.59•	3.011±2.98•	0.027±0.35	4.797±2.59•	5.146±2.70•	2.252±0.46	31.23±12.9•	32.63±13.2•	13.19±3.50
Overall	0.863±0.59•	0.563±0.44•	-0.11±0.22	2.832±0.83•	2.146±0.70•	1.430±0.20	13.75±4.72•	12.39±4.06•	7.300±0.67

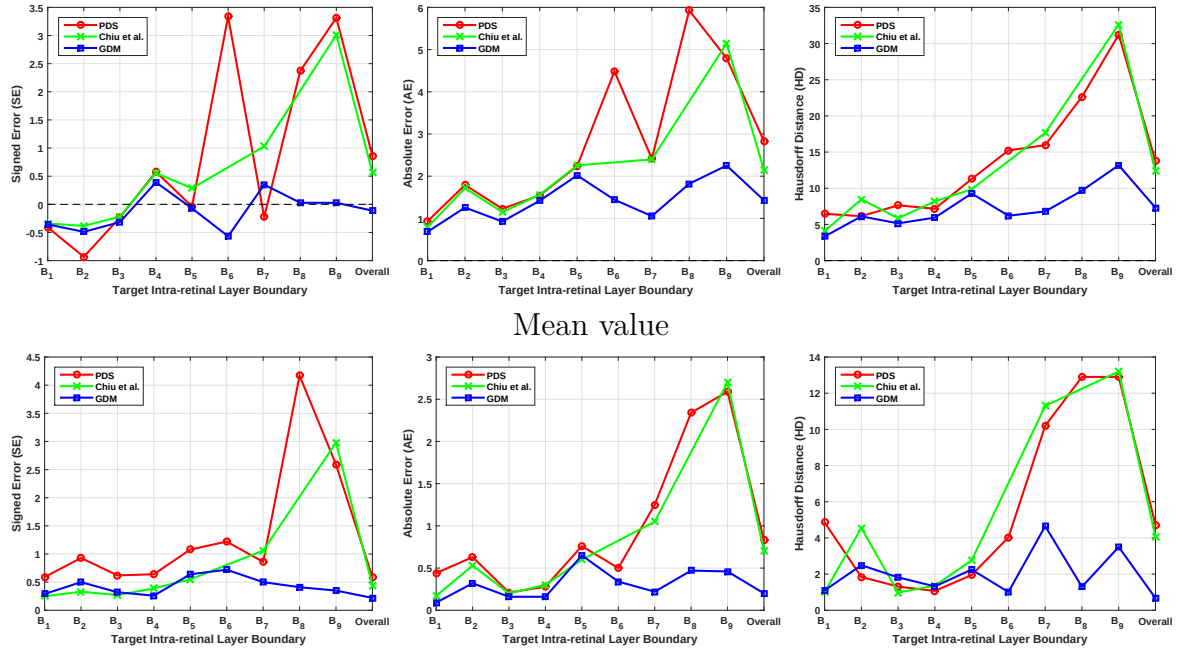


Figure 4.12: Plots of mean and standard derivation obtained by different methods in Table 4.4 for pathological B-scans. The 1st and 2nd rows respectively denote the mean and standard derivation of the SE (μm), AE (μm) and HD (μm) for segmenting boundaries $B_1 - B_9$ using PDS, Chiu's mehtod and GDM. The overall value is the average result over all boundaries.

In Table 4.3 and Figure 4.11, the SE values show that PDS leads to large segmentation bias with the largest error being $7.45\mu m$, whilst GDM results in small bias with the largest error being $0.92\mu m$. The mean SE plot of GDM is close to zero, meaning that GDM is less biased than PDS and Chiu's method. Large errors from PDS normally take place at

B_1, B_2, B_6 and B_8, which is consistent with visual inspection on the segmentation results of healthy B-scans in Figure 4.10. Furthermore, the AE values show that GDM performs better for most of the segmented boundaries. Particularly at B_1 and B_2 where the curved fovea region is located, the HD values from GDM $\{3.702 \pm 1.62\mu m, 7.340 \pm 2.16\mu m\}$ are significantly lower than those from PDS $\{36.56 \pm 15.9\mu m, 29.00 \pm 11.6\mu m\}$ and Chiu's method $\{22.12 \pm 9.23\mu m, 21.25 \pm 5.98\mu m\}$. However, the accuracy of different methods are comparable at flat or smooth retinal boundaries such as B_4, B_7 and B_9. Finally, since the manual segmentation traces small bumps of the true boundaries and the segmentation results by PDS are however very smooth, the overall accuracy of PDS is the lowest among all the approaches compared.

In Table 4.4 and Figure 4.12, we can see that GDM is more accurate and robust compared with the other two methods for pathological data. Larger errors have been found at the last four boundaries B_6, B_7, B_8 and B_9 for all the segmentation methods. This is because the dry age-related macular degeneration has led irregularities to these retinal boundaries, making them less accurate and robust. The overall accuracy measured by the three quantities (SE, AE and HD) has also decreased compared with the counterparts listed in Table 4.3. Chiu's method using the Dijkstra's algorithm can be deemed as a discrete approximation of the proposed GDM. Therefore, its results are comparable to the GDM results at some flat retinal boundaries while much better than the PDS results. However, the fast sweeping algorithm used to solve the Eikonal equation guarantees local resolution for the geodesic distance, which reduces grid bias significantly and attains sub-pixel accuracy for the geodesic path result from GDM. In addition to the novel weight function proposed in (4.4), GDM also resolves the metrication problem caused by discrete graph methods and thus can achieve more accurate results than Chiu's method.

In the next section, GDM is used to segment OCT volume dataset that includes samples from ten healthy adult subjects, named as Volume 1 to 10 respectively. Dufour's method and OCTRIMA3D are also used to segment the same dataset for comparison purposes. In Figure 4.13, we demonstrate four representative segmentation results using GDM on Volume 1, 2, 7 and 9.

Two representative B-scans in Volume 4 are shown in Figure 4.14 and their segmentation results by the three approaches are shown in Figure 4.15. Note that one B-scan retinal structures are quite flat and the other contains the nonflat fovea region. Dufour's method has lower accuracy than OCTIMA3D and GDM for both cases. OCTRIMA3D extends Chiu's method to 3D space and improves it by reducing the curvature in the fovea region using the inter-frame flattening technique, so the method performs very well for both flat and nonflat retinal structures. However, there are still some errors at

B_5. OCTRIMA3D is able to flatten B_1 and in the meanwhile it also increases curvature of its adjacent boundaries such as B_5, which might be the reason leading to the errors. Compared with the other two, GDM results show less green lines, verifying that the results are closer to ground truth and thus it is the most accurate among the three compared. In addition to the 2D visualisation, the 3D rendering of the results segmented by the three approaches on Volume 4 is given in Figure 4.16. The experiment furthermore shows that Dufour's results deviate more from ground truth, while OCTRIMA3D is better than Dufour's method and is comparable to GDM. GDM results cover less grey ground truth and are the best.

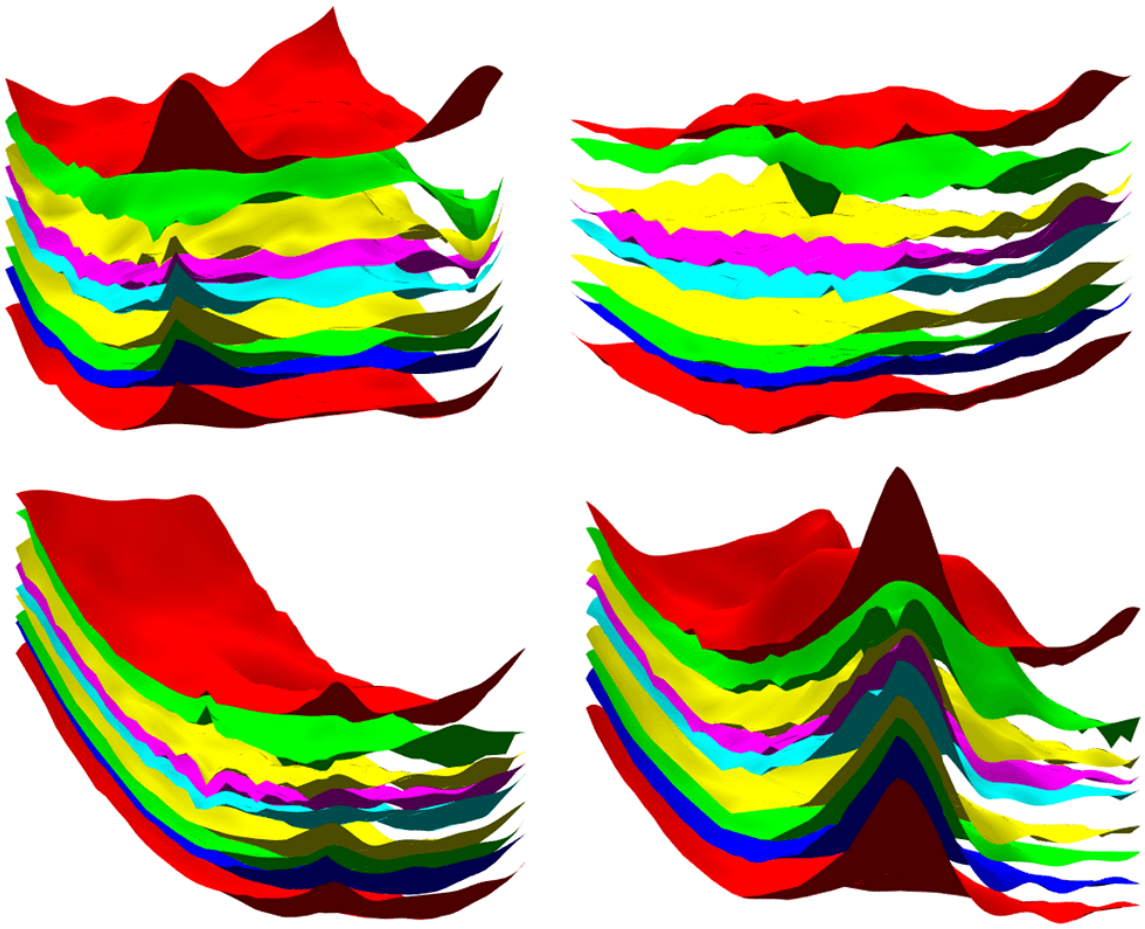


Figure 4.13: 3D rendered images of human in vivo intra-retinal layers obtained through segmenting 3D SD-OCT images with the proposed GDM method. Samples are named Volume 1, Volume 2, Volume 7 and Volume 9. The colour used for each individual retinal layer is the same as in Figure 4.2.

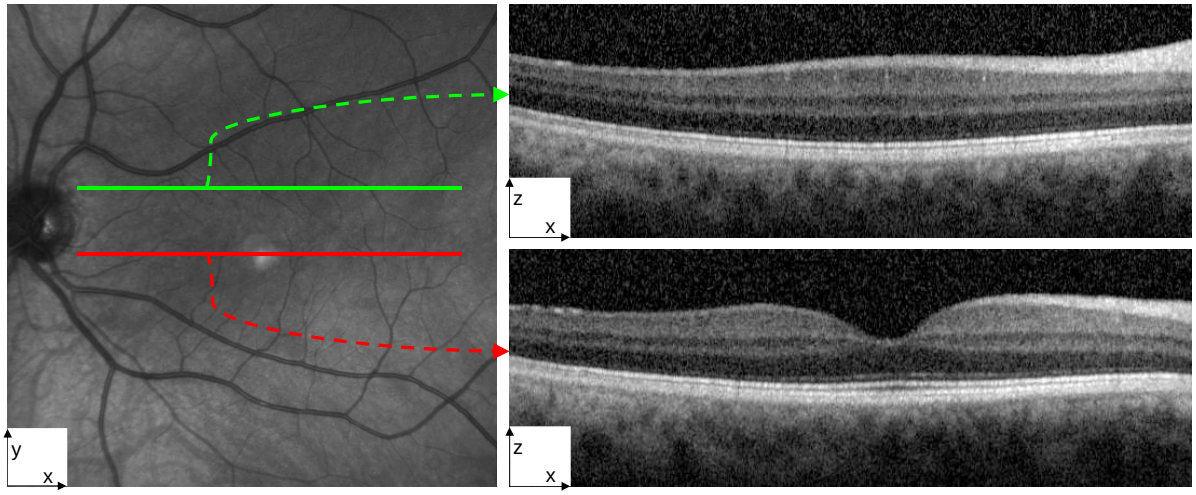


Figure 4.14: Two B-scans extracted from Volume 4. The left shows the en-face representation of the OCT scan with the overlaid green and red lines representing the corresponding two B-scans in the right.

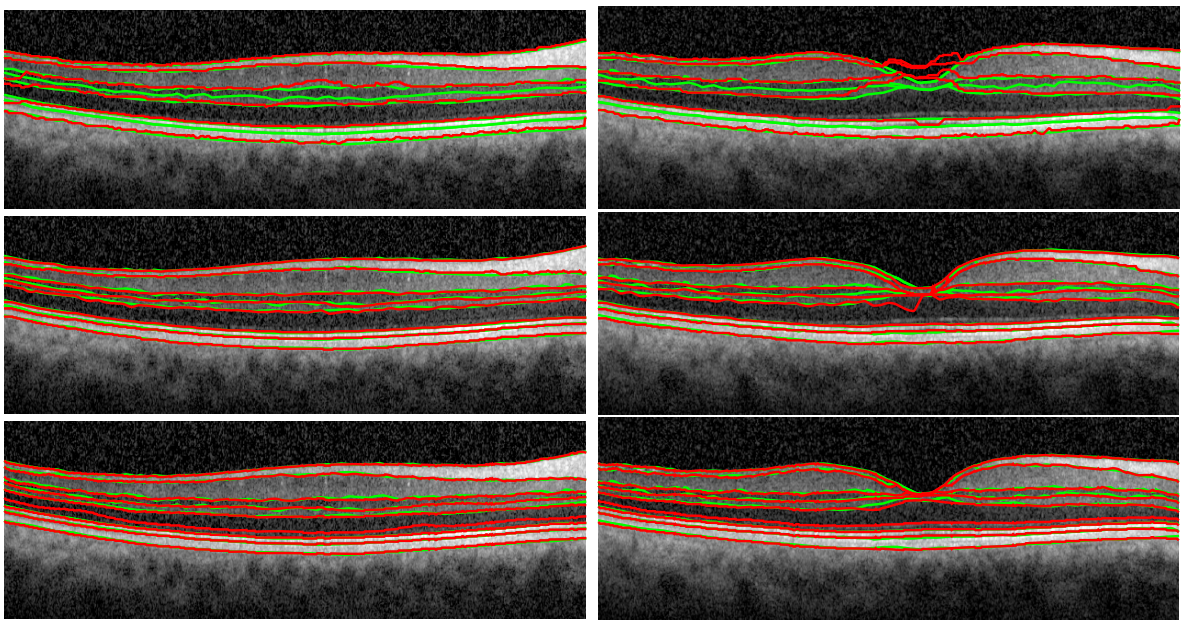
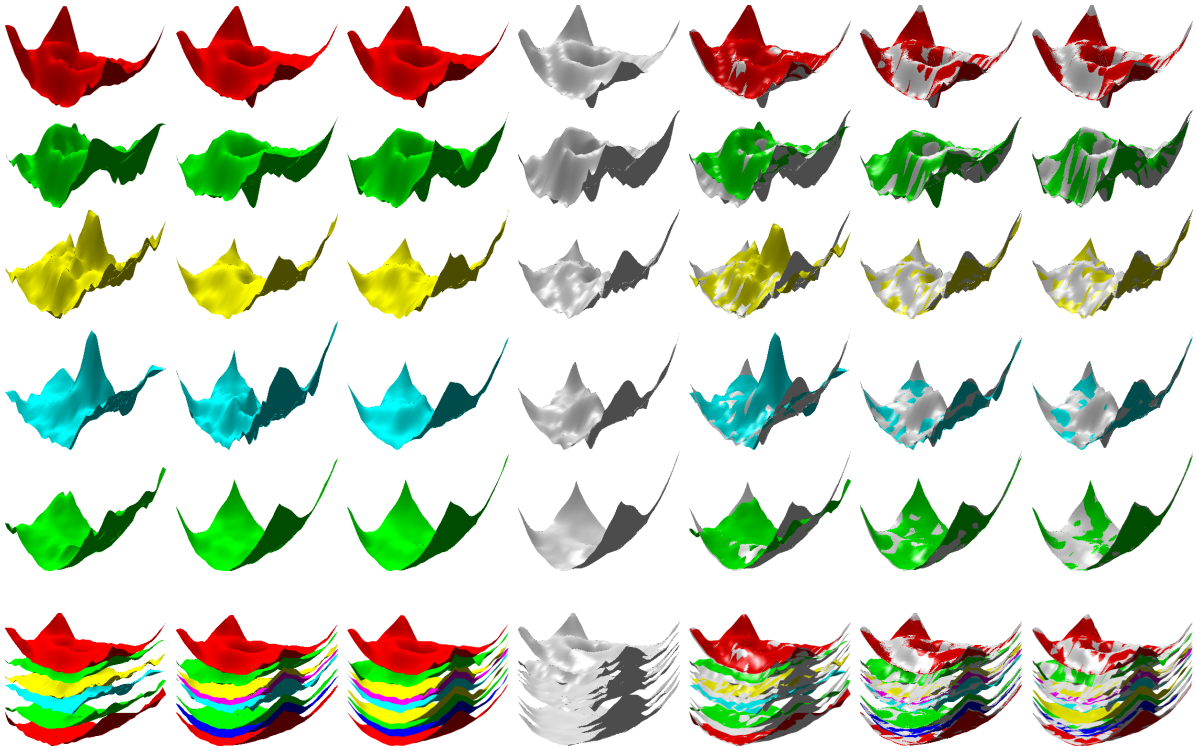


Figure 4.15: Comparison between Dufour's method (left), OCTRIMA3D (middle) and GDM (right) on the two B-scans in Figure 4.14. The segmentation lines by these methods are marked with red lines while the manual labelled ground truth with green lines.



(a) (b) (c) (d) (e) (f) (g)

Figure 4.16: 3D comparison between Dufour's method, OCTRIMA3D and GDM by segmenting the intra-retinal layers from Volume 4. Column (a)-(d) are respectively Dufour's results, OCTRIMA3D results, GDM results and ground truth. Column (e)-(g) are respectively the segmentation results of the three methods, overlaid with ground truth. Row 1-6 represent the B_1, B_2, B_3, B_5, B_7 and overall retinal layer surfaces, respectively.

Table 4.5: Comparison of SE (μm), AE (μm) and HD (μm) calculated using the results of different methods (Dufour's method, OCTRMIA3D and GDM) and manually segmented ground truth, for the OPL-ONL (B_5) layer surface in each of 10 OCT volumes.

	SE (μm)			AE (μm)			HD (μm)		
Volume #	Dufour et al.	OCTRIMA3D	GDM	Dufour et al.	OCTRIMA3D	GDM	Dufour et al.	OCTRIMA3D	GDM
1	-1.194	0.4559	0.3782	2.3816	1.3490	1.0720	25.688	15.273	10.449
2	-2.170	-0.036	-0.128	4.5250	0.9089	0.7814	56.667	11.570	7.0938
3	-2.576	0.4182	0.5983	3.6129	1.3237	1.0989	25.203	16.719	9.5326
4	-2.296	1.0987	0.6774	3.8185	1.5175	1.0753	51.522	18.364	9.6151
5	-1.680	1.3288	0.5909	4.3327	1.5012	0.9005	56.223	11.889	8.8419
6	-2.623	1.0732	0.2974	4.0682	1.4838	0.9493	43.070	19.201	9.5281
7	-2.326	0.5294	0.4529	3.1506	0.9378	0.7433	31.782	8.6701	6.4803
8	-0.636	1.1355	0.6833	2.3955	1.4455	1.0069	25.481	17.930	11.685
9	-4.206	0.3077	0.0859	4.5813	1.0780	0.7678	43.223	8.9694	5.7191
10	-2.648	0.6701	0.2606	4.4903	1.0627	0.7877	41.017	11.666	10.961

Table 4.6: Comparison of SE (μm), AE (μm) and HD (μm) calculated using the results of different methods (Dufour's method, OCTRMIA3D and GDM) and manually segmented ground truth, for the IS-OS (B_7) layer surface in each of 10 OCT volumes

	SE (μm)			AE (μm)			HD (μm)		
Volume #	Dufour et al.	OCTRIMA3D	GDM	Dufour et al.	OCTRIMA3D	GDM	Dufour et al.	OCTRIMA3D	GDM
1	-0.432	-0.148	-0.019	1.1013	0.5391	0.4437	16.559	4.7616	4.5805
2	0.7476	-0.276	-0.079	2.0329	0.5539	0.3971	20.309	5.2093	3.7743
3	-0.311	-0.291	-0.106	1.4347	0.5406	0.4629	18.432	2.9790	4.0176
4	0.3652	-0.116	0.3363	1.6954	0.5271	0.4601	27.853	5.3672	2.7882
5	0.6057	-0.098	0.0994	1.7567	0.4756	0.3500	26.556	3.7573	3.4150
6	0.9825	-0.592	-0.139	2.4970	0.7247	0.4066	23.487	5.9301	3.9297
7	-1.247	-0.536	0.0237	1.3895	0.7501	0.3716	10.016	3.1398	3.6980
8	-0.311	-0.069	0.1740	1.0438	0.4053	0.3466	15.044	4.2301	4.3940
9	-0.755	-0.111	0.1407	0.8068	0.5422	0.3939	3.5210	3.4263	3.3868
10	-0.099	-0.220	0.1028	1.2941	0.5609	0.4246	13.313	3.1210	3.5361

Table 4.7: Comparison of OSE (μm), OAE (μm) and OHD (μm) calculated from the results of different methods (Dufour's method, OCTRMIA3D and GDM) and manually segmented ground truth, for the overall retinal layer surfaces in each of 10 OCT volumes

	OSE (μm)			OAE (μm)			OHD (μm)		
Volume #	Dufour et al.	OCTRIMA3D	GDM	Dufour et al.	OCTRIMA3D	GDM	Dufour et al.	OCTRIMA3D	GDM
1	-1.271	0.3607	0.4338	1.8358	1.1204	0.9538	17.486	9.3358	7.9163
2	-1.161	0.0246	0.0640	2.5380	0.9652	0.7238	29.682	7.7987	6.1267
3	-1.513	-0.052	0.3456	2.1470	0.9343	0.7838	19.985	8.3491	6.9920
4	-1.431	0.4272	0.3560	2.5278	1.0374	0.8667	31.346	9.4042	7.3130
5	-1.020	0.6369	0.5021	2.4119	1.0794	0.8289	32.607	8.6822	7.1379
6	-1.434	0.4216	0.3969	2.6754	1.1371	0.8606	28.629	9.5267	7.2548
7	-2.010	0.0059	0.3283	2.2458	0.9682	0.7407	21.788	7.0644	6.8279
8	-1.031	0.5815	0.5785	1.7462	1.1063	0.9067	17.610	10.100	8.5112
9	-1.951	0.0542	0.2014	2.1368	0.8771	0.6922	21.344	5.7482	5.4794
10	-1.513	0.1022	0.2109	2.3315	0.8397	0.6596	24.841	6.3250	6.7132

Table 4.5-4.7 contain the quantitative accuracy comparison of the three methods on 10 OCT volumes. Table 4.5 shows the results for layer boundary B_5 around the fovea region, while Table 4.6 presents the results for boundary B_7 which is flatter and smoother. In Table 4.5, the SE values indicate that Dufour's method produces larger segmentation bias than OCTRIMA3D and GDM. The SE values by GDM are in the range of [-0.128μm

$0.6833\mu m$], showing less variability than those by the other two methods. Moreover, GDM leads to the smallest AE and HD values in all 10 cases, indicating that GDM is the most accurate among all the methods. Compared with Table 4.5, Table 4.6 shows a significant improvement of all the methods. For example, the range of the HD values by Dufour's method has dropped from [$25.688\mu m$ $56.667\mu m$] to [$3.521\mu m$ $27.853\mu m$]. In addition, the accuracy gap between OCTRIMA3D and GDM has been reduced, and in Volume 3, 7 and 10 the HD values by OCTRIMA have even become smaller than those by GDM. These improvements are due to the fact that the retinal layer boundary B_7 is flatter and smoother than B_5. From the values of OAE and OHD in Table 4.7, we observe that the accuracy of GDM is the highest for the segmentation of total retinal boundaries from each of 10 OCT volume.

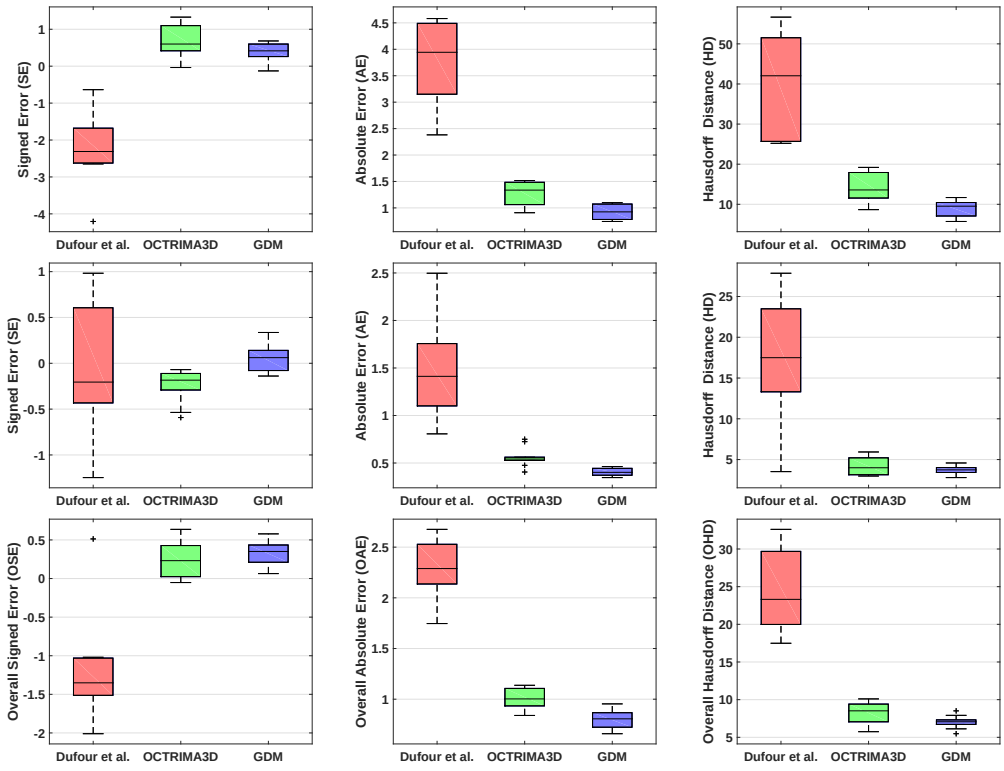


Figure 4.17: Boxplots for SE (μm), AE (μm), HD (μm), OSE (μm), OAE (μm) and OHD (μm) obtained by different methods in Table 4.5-4.7 for 10 OCT volumes. 1st row: boxplots of Table 4.5; 2nd row: boxplots of Table 4.6; 3rd row: boxplots of Table 4.7.

The corresponding boxplots of Table 4.5-4.7 are shown in Figure 4.17. These boxplots show that the proposed GDM method performs consistently better, with higher accuracy and lower error rates for both flat and nonflat retina layers. There is little variation in performance across different structures and even in the worst case scenario the proposed method yields lower error rate than the average performance of other methods. In

Figure 4.18 we present 3D plots of the SE, AE and HD values computed by the three methods on the 10 volumes. For GDM, its SE values are closer to zero and its AE and HD values remain smaller. The overall distribution of these data points also indicates that the GDM results are less oscillating. We can thus conclude that GDM performs the best among all the methods compared for extracting intra-retinal layer layers from 3D OCT volumes.

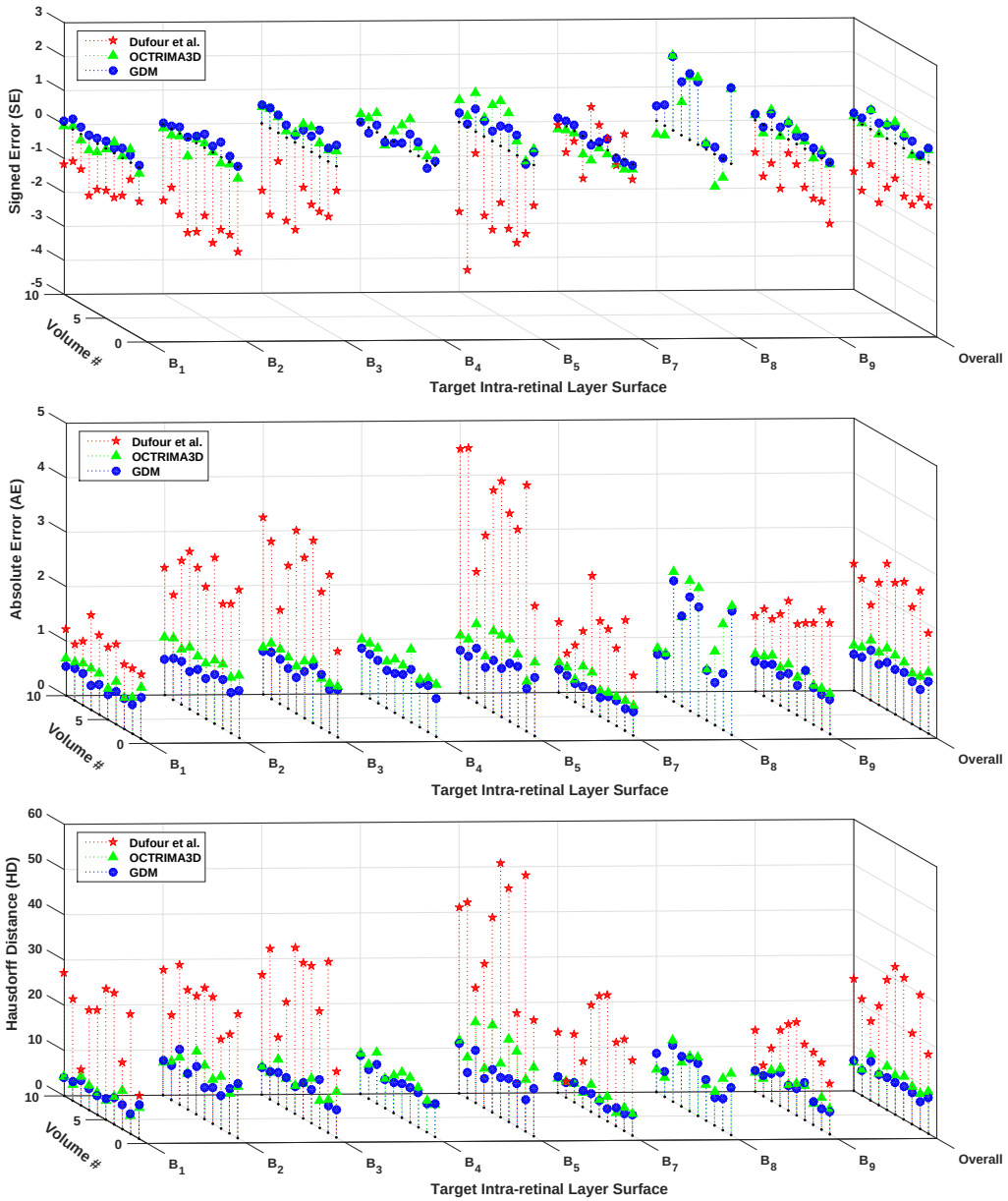


Figure 4.18: 3D plots of SE (μm), AE (μm) and HD (μm) obtained using Dufour' method, OCTRMIA3D and GDM on 10 OCT volumes.

4.4.5 Computational time

In this section the performance of the different approaches in terms of the computation time is demonstrated. We implemented PDS, Chiu's method and GDM using Matlab 2014b on a Windows 7 platform with an Intel Xeon CPU E5-1620 at 3.70GHz and 32GB memory. For a 633×496 sized B-scan, with initialisation close to the true retinal boundaries, it takes 3.625s (500 iterations) for PDS to detect two parallel boundaries. Chiu's method needs 1.962s to detect one layer boundary, while GDM only takes 0.415s. Note that the time complexity of Chiu's graph search method is $O(|E|log(|V|))$, where $|V|$ and $|E|$ are the number of nodes and edges and $|V| = MN$ and $|E| = 8MN$ in boundary detection in an image. Hence the time complexity of the method is $O(MNlog(MN))$. In contrast, our GDM solved using fast sweeping has linear complexity of $O(MN)$, which is more efficient than Chiu's method. For 3D segmentation, OCTRMIA3D explores spatial dependency between two adjacent B-scans and applies Chiu's method to each 2D slice independently. OCTRMIA3D is thus able to track retinal boundaries in 3D OCT images efficiently. It was reported in [155] that the processing time of the OCTRMIA3D for the whole OCT volume of $496 \times 644 \times 51$ voxels was 26.15s, which is faster than our GDM (40.25s is used to segment a $496 \times 633 \times 10$ sized volume). Finally, Dufour's graph method needs 14.68s to detect the six intra-retinal layer boundaries on a $496 \times 633 \times 10$ sized volume. Dufour's method was implemented using a different programming language (C) and it detected different number of retinal layers from that of GDM, so comparison cannot be made between the two methods.

4.5 Summary

In this chapter, we presented a new automated retinal layer segmentation method based on the geodesic distance for both 2D and 3D OCT images. The method integrates horizontal and vertical gradient information and can thus account for intensity changes in the both directions. Furthermore, the exponential weight function employed within the approach enhances the foveal depression regions and weak retinal layer boundaries. As a result, the proposed method is able to segment complex retinal structures with large curvatures and other irregularities caused by pathologies. Extensive numerical results, validated with ground truth, demonstrate the effectiveness of proposed method for segmenting both normal and pathological OCT images. The proposed method has achieved higher segmentation accuracy than the state of the art methods compared, such as the parametrised active contour model and the graph theoretic based approaches.

Ongoing research includes integrating the segmentation framework into a system for detection and quantification of retinal fractures and other eye diseases.

Chapter 5

Surface reconstruction from point clouds

5.1 Introduction

Reconstructing an accurate and smooth 3D surface from a point cloud is a challenging problem as the point cloud consists of unorganised and unconnected points. In general, there are two approaches to represent a surface: explicit or implicit. Explicit representation [165, 166] describes the location of points as well as the local geometry of a surface explicitly. It can be accurate and efficient, but less robust and less flexible in handling arbitrary and dynamically changing surface topology. Implicit representation [2, 167, 168] usually constructs a surface as an evolving level set function [169], starting with an initial surface. This approach is topologically flexible and robust in dealing with noisy and non-uniform point clouds [167, 168].

One of the most successful implicit surface reconstruction methods based on the variational level set method was proposed by Zhao et al. [2, 170]. The model however has some drawbacks. First, periodical reinitialisation is needed to keep the evolving level set close to a signed distance function to maintain stable surface evolution. This is a tedious and expensive procedure and may even cause the surface to shrink [171]. Second, the model is non-convex, and the reconstruction result is sensitive to the initial condition. In addition, if the point cloud contains fine details or concave regions, the evolving level set surface often gets stuck in a local minimum even if the initial surface is very close to the true surface.

To overcome the problems with existing implicit reconstruction methods, in this chapter we introduce a novel variational model for reconstructing accurate surfaces from point clouds. We transform the surface reconstruction problem into an implicit

image segmentation problem. To do so, we will need to create an edge indicator, an enclosed image and an initial surface directly from the given point cloud. The edge indicator function provides information about the precise location of the surface (to be reconstructed). The enclosed image provides a indication of a grid point to be inside or outside of the surface. The initial surface provides a good initialisation for the level set function used in the proposed model.

There are two main steps in our reconstruction process. Step 1: a distance function is derived from a point cloud using the fast sweeping algorithm. It is then used: 1) as an edge indicator function, 2) to calculate an image (2D or 3D) enclosed by the point cloud, and 3) to find an initial surface. Step 2: this initial surface is then evolved by the proposed variational level set model, which effectively integrates the edge indicator and the image derived from Step 1. Gradient descent optimisation is then adopted to minimise the proposed variational model and thereby accurately reconstruct the surface from a given point cloud. Experiments show that the proposed method is more accurate than state-of-the-art methods including the Poisson method [172]. The two reconstruction steps are detailed in the following sections.

5.2 The initial image and the edge indicator

5.2.1 Calculation of distance function using fast sweeping

One of the earliest successful implicit surface reconstruction methods was developed by Zhao et al. [2, 170]. Assume we have a unorganised point cloud $\{x_i\}$, where $x \in \Omega$ and $i = 1, ..., N$ (N is the number of the point cloud). The reconstructed surface/curve is obtained by minimising the weighted geodesic active contour (GAC) energy

$$E(\Gamma) = \int_{\Gamma} d(x) d\Gamma, \tag{5.1}$$

where Γ is a curve in $\mathbb{R}^2$ and a surface in $\mathbb{R}^3$. $d\Gamma$ is the arc length element or the surface parametrisation. The variational method is known as the GAC model or snake model, which was first proposed in [173] and extended to the level set framework by Caselles, Kimmel, and Sapiro [174]. $d(x)$ in (5.1) is the distance function of the point cloud, which satisfies the following Eikonal equation

$$|\nabla d(x)| = W(x), \ \forall x \in \Omega \backslash \{x_i\}, \tag{5.2}$$

where

$$W(x) = 1$$

and

$$d(x_i) = 0,\ \forall x \in \{x_i\}\,.$$

Appendix B shows the detailed implementation of (5.2) using fast sweeping [11, 175] in both 2D and 3D. Figure 5.1 (c) and (d) show the distance maps calculated from the point clouds (a) and (b) using fast sweeping. The distance $d(x)$ is further used as an edge indicator function to derive an initial image enclosed by the point cloud.

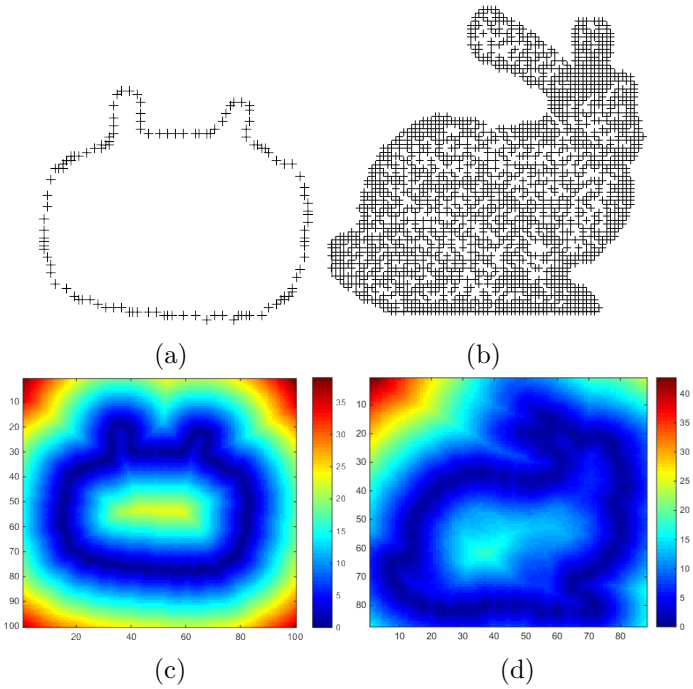


Figure 5.1: Calculating the distance function for the original point cloud. (a)-(b) are 2D and 3D point clouds respectively. (c) is the distance function for (a). (d) is a cross sectional view of the distance function for (b).

5.2.2 Initial image calculation using the distance function

In [167], the authors calculate the image enclosed by a point cloud using a distance function $d(x)$ calculated from the point cloud. Suppose an image $f(x)$ is given, then a natural choice of an edge indicator function for $f(x)$

$$g\left(x\right) = \frac{1}{\varepsilon + |\nabla f(x)|^p},$$

where p and ε are positive parameters, and g is close to zero at edge locations. The next step is to view the distance function $d(x)$ as the edge indicator, i.e., $g(x) = d(x)$. By regularising the distance function through $g(x) = d(x) + \varepsilon$. The initial image $f(x)$ is calculated by solving the following Eikonal equation

$$|\nabla f(x)| = \frac{1}{\varepsilon + g(x)^{1/p}}.$$

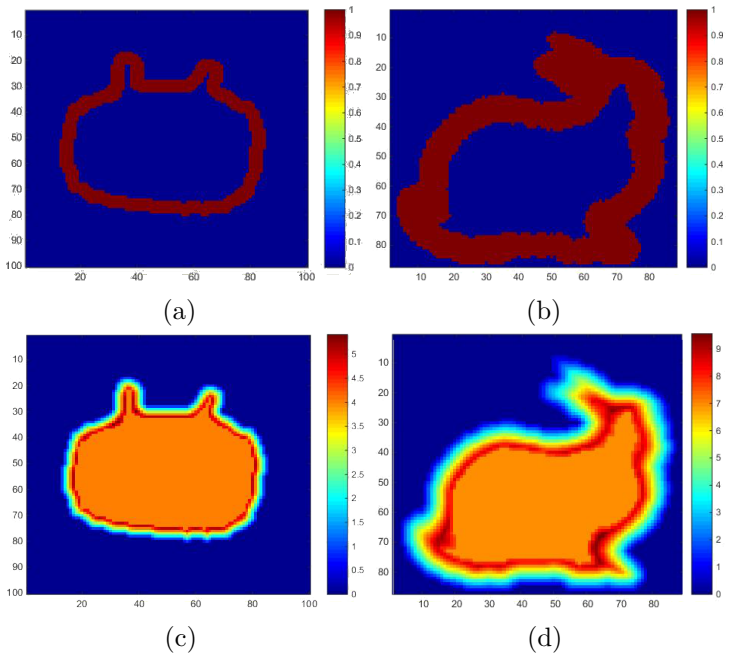


Figure 5.2: Finding enclosed image from annular binary images. (a) and (b) are two annular binary images. (c) and (d) are the images calculated from (a) and (b) respectively. (b) and (d) are cross sections of the corresponding images.

In [168], the authors used the inner product field to give a good indication of interior and exterior regions of a point cloud. The underlying idea to computing an image $f(x)$ is given by

$$f(x) = (x - cp(x)) \cdot n(cp(x)).$$

At a point x in the domain Ω, $cp(x)$ denotes a point on the point cloud that is closest to x, and $n(cp(x))$ is the outward normal at $cp(x)$. The calculated $f(x)$ is positive when x is outside the point cloud and negative when x is inside the point cloud.

We propose a simpler method to compute the initial image enclosed by the point cloud, which is different from the above two methods. Specifically, we first threshold the distance function $d(x)$ in (5.2) to get an annular binary image $I(x)$, as shown in the first two images in Figure 5.2. The fast sweeping algorithm is then used on $I(x)$ to find the

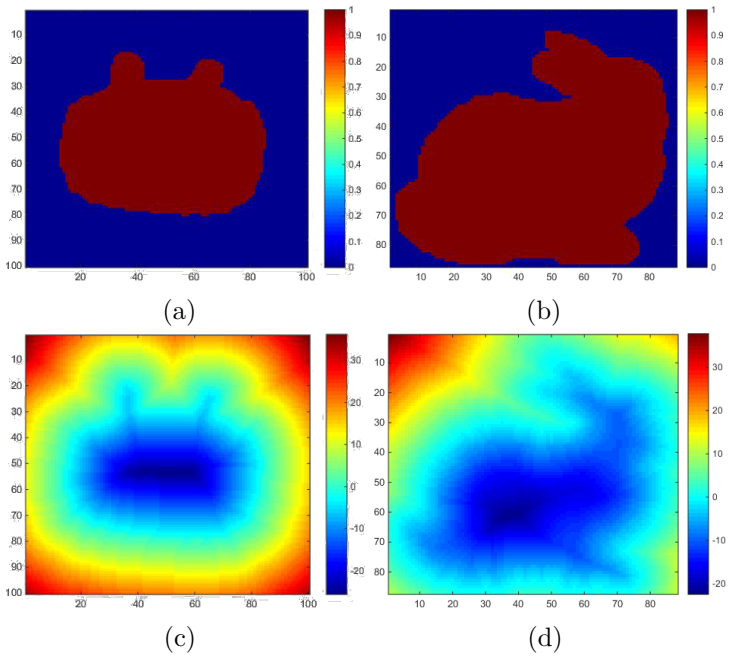


Figure 5.3: Calculate a signed distance function from the binary image. (a) and (b) are two binary images. (c) and (d) are corresponding signed distance maps of (a) and (b) respectively. (b) and (d) stand for one slice of the 3D data.

image. Mathematically, $W(x) = I(x)$ replaces $W(x) = 1$ on the right-hand side of (5.2), rendering $d(x)$ no longer a distance function. To solve $d(x)$ using fast sweeping with eight directional sweeping Gauss-Seidel iterations, the grid points on the 6 boundary faces in the domain Ω are set to zero and other interior grid points are set to very large values. Once $d(x)$ are found, and the image is set to $f(x) = d(x)$, as shown in the last two images in Figure 5.2. $f(x)$ is then used to find a good initial surface to speed up the level set evolution in Section 5.5. $f(x)$ is first thresholded to obtain a new binary image shown in the first two images in Figure 5.3. A simple algorithm (such as March Cube) can be employed to find all the points $\{\widetilde{x}_i\}$ on the boundary of the object in the binary image. These new points $\{\widetilde{x}_i\}$ are very close to the original unorganised point cloud $\{x_i\}$. The fast sweeping algorithm is applied again to the new point cloud $\{\widetilde{x}_i\}$ to obtain a signed distance function ϕ_0 with the sign information obtained from the binary image (i.e., inside is negative and outside is positive). ϕ_0 now is a good initialisation for the proposed method in Section 5.5.

5.3 Mathematical representation of implicit surfaces

After the initial image $f(x)$ and distance function $d(x)$ are obtained from the point cloud, the implicit surface reconstruction problem can be posed as a variational image segmentation problem. We will first introduce how to mathematically define an implicit closed curve C (surface in 3D) and its corresponding arc length term. The theory can be easily extended to higher dimensions. A new variational level set segmentation model, that use the arc length term defined by an implicit contour, will be proposed subsequently for surface reconstruction.

5.3.1 Implicit contours

Implicit contours are contours of the form

$$C = \{x \in \Omega \mid \phi(x) = 0\}. \tag{5.3}$$

which means that the contour C is given by the zero level set of a function $\phi : \Omega \to \mathbb{R}$

A number of well-known segmentation methods have been formulated in terms of implicit contours. The initial contour is embedded in a surface, for example by the signed distance function. The contour evolution is replaced by an evolution of the embedding surface, and the corresponding contour at a given time is obtained by determining the zero level set of the evolving surface.

Implicit contour representation has several advantages and disadvantages:

+ The evolving contours can be automatically adapted to topological changes (i.e., contours can be merging or spiting), as shown in Figure 5.4.

+ The evolution of level set function is on fixed grids on which the grid points never collide or drift apart. This guarantees the stability of contour evolution as well as permits the usage of high efficient numerical scheme.

− As contours are expressed implicitly, the computational space is extended from two dimension to three dimension. The computational cost thus becomes much more expensive.

− Without touching the image boundaries, single level set function can only represent closed curves in the image. To define open curves, at least two level set functions are required, which further reduces the computational speed.

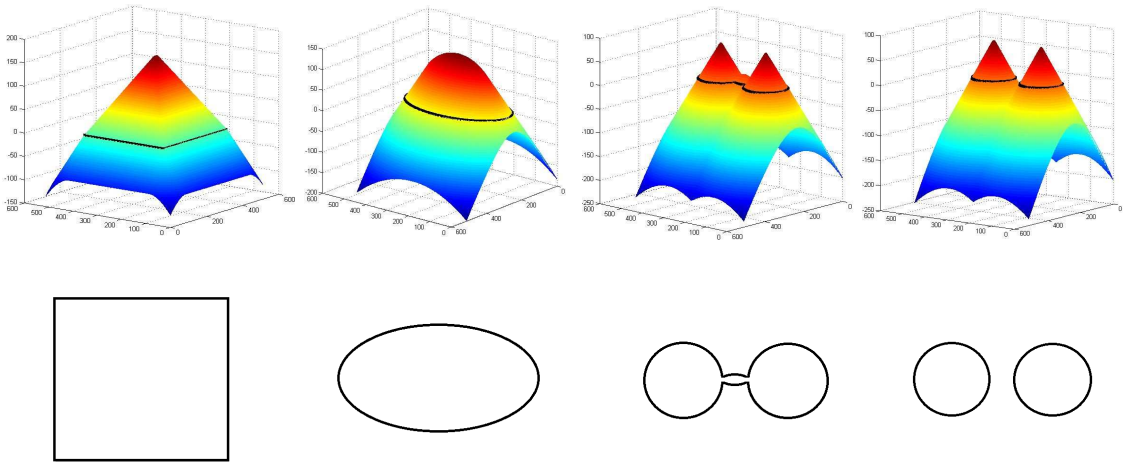


Figure 5.4: Level set evolution. 1st row: 3D level set functions with their zero level sets marked in black. 2nd row: zero level set contours. This example shows level set methods can automatically adapt to topological changes (i.e., contours merging or spiting).

5.3.2 Length of implicit contours

So far, we have introduced how to represent a moving contour with an implicit expression. It is yet remaining unclear that what the explicit mathematical expression of the length of a curve is. Therefore, it is infeasible to derive the Euler-Lagrange equations of the variational formulations and apply a straightforward numerical scheme to implement them. Technically, to implicitly define the length of a contour, it often requires the introduction of the Heaviside and Dirac functions, which are defined as

$$H(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad \text{and} \quad \delta(x) = \frac{d}{dx} H(x), \tag{5.4}$$

where the Dirac function is the derivative of the Heaviside function in the distributional sense. According to the coarea formula of a binary function, we have

$$|C| = \int_{\Omega} |\nabla H(\phi)| dx = \int_{\Omega} |\nabla \phi| \, \delta(\phi) dx. \tag{5.5}$$

Here $\Omega \subset \mathbb{R}^n$ and n=2 and 3 receptively denote 2D image and 3D image spaces. Function ϕ normally stands for the signed distance function, a commonly used level set function. Note that by applying formulation (5.5), the original length $|C|$ has been converted to a higher dimensional space Ω. As (5.5) combines both the level set method and variational framework, energy functionals that are utilising such term is normally referred to as variational level set methods [119–123, 65].

Nevertheless, the respective energies that involve (5.5) are not convex and can only determine local optima, meaning that different initialisations will lead to different segmentation results. Since 2005, researchers have proposed novel variational approaches which aim to approximate the original energies with convex functionals. Rather than minimising the original energy locally, they minimise an approximation of the original energy globally. For a two-region image segmentation problem, this can be done by using following convex length term instead of (5.5)

$$|C| = \int_{\Omega} |\nabla u| dx, \tag{5.6}$$

where $u : \Omega \to \{0, 1\}$, $\Omega = \Omega_1 \cup \Omega_2$, $u(x) = 1$ if $x \in \Omega_1$ and $u(x) = 0$ if $x \in \Omega_2$. Chan et al. [176] then employ the relaxation technique by simply dropping the constraint that u must be binary. They allow u to take on values in the entire interval $[0, 1]$, which is the convex hull of the original domain $\{0, 1\}$. Afterwards, the problem becomes a convex relaxation optimisation problem, which guarantees a global optimum and can be solved by more sophisticated convexification algorithms in addition to the gradient descent flow.

5.4 Convexified variational methods for image segmentation

In this section, we introduce some state-of-the-art convexified variational image segmentation methods that will be compared against the proposed method for the purpose of implicit surface reconstruction.

5.4.1 Convex relaxation model based on binary function

One popular segmentation method using (5.6) as an implicit length term is as follows

$$\min_{u \in [0,1], c_1, c_2} \left\{ \int_{\Omega} |\nabla u| dx + \lambda \int_{\Omega} Q(c_1, c_2) u dx \right\}. \tag{5.7}$$

Above $Q(c_1, c_2) = (c_1 - f(x))^2 - (c_2 - f(x))^2$, where $f(x)$ is a given image that is not necessarily binary and c_1, c_2 are two constants. The above model is a convex constrained version of the original Chan-Vese model [177], because the relaxation technique is employed to relax u to $[0, 1]$. The functional in (5.7) is convex in u and (c_1, c_2), separately. However, it is not biconvex. In fact, (5.7) coincides with the piecewise constant Mumford-Shah segmentation model. In [126], the authors modified the problem to incorporate

information from an edge indicator function and proposed the weighted total variation (TV) regularisation. Their method has the form of

$$\min_{u\in[0,1],c_1,c_2} \left\{ \int_\Omega g\left(x\right)|\nabla u|dx + \lambda \int_\Omega Q\left(c_1,c_2\right)udx \right\}, \tag{5.8}$$

where $g(x)$ is an edge indicator function, which is equivalent to the distance function $d(x)$ in the surface reconstruction problem. In fact, the weighted TV term in (5.7) alone is the geodesic active contour model (5.1) with convex binary length approximation. The proof of convexity of (5.4.2) with respect to u can be found in Theorem 3 in [126]. This weighted term identifies objects using the edge indicator function, which takes small values near boundaries (where the image gradient is large) and large values where the image is smooth. Analogously, in surface reconstruction this term is small near or on the scattered points, and becomes increasingly large further away from the point cloud.

Once this optimisation problem is solved, the reconstructed surface is found by thresholding the level set function to get

$$\Omega = \{x : u\left(x\right) > \alpha\} \tag{5.9}$$

for some $\alpha \in (0,1)$.

In summary, (5.7) approach for surface reconstruction proceeds as follows

1: **while** "not converged" **do**
2: Define $Q^k = (c_1^k - f)^2 - (c_2^k - f)^2$
3: Solve $u^k = \min_{0\le u\le 1} \left\{ \int_\Omega^k g\left(x\right)|\nabla u|dx + \lambda \int_\Omega Qudx \right\}$
4: Find $\Omega^k = x : u^k(x) > \alpha$
5: Update $c_1^{k+1} = \int_{\Omega^k} fdx$ and $c_2^{k+1} = \int_{(\Omega^k)^c} fdx$
6: **end while**

The bottleneck of this segmentation algorithm is the computation of the minimiser of step 3 above. The split Bregman approach, which has been introduced to minimise the different variational denoising model in Chapter 2, can be adopted to minimise this problem. Such method has the advantage of being a more efficient solver for step 3.

5.4.2 Convex formulation based on mean curvature motion

In this section, we show how the GAC/snake model (5.1) can be well approximated using the convex TV functional. Conventional techniques for evolving the GAC contour rely

on the following explicit discretisations of the gradient flow

$$\partial_t C = (d\kappa - \langle \nabla d, \mathcal{N} \rangle) \mathcal{N}. \tag{5.10}$$

Equivalently to (5.10), in [178] Chambolle introduced an implicit scheme that evolves the contour using a sequence of convex variational problems involving the TV functional. The derivation presented here follows the approach presented in [179, 180], which adapts Chambolle's mean curvature motion to the GAC energy. Given an initial curve C, let s_C be its representation as a signed distance function and now consider the following convex minimisation problem

$$\phi^* = \arg\min_{\phi} \left\{ \int_\Omega d\,(x)\, |\nabla \phi| dx + \frac{1}{2h} \int_\Omega (\phi - s_C)^2 dx \right\}. \tag{5.11}$$

Again we recall $d(x)$ here is the distance function computed from those discrete points, which is equivalent to an edge indicator. Now assume the new curve C' is the zero level set of the level set function $\phi^*(x)$, that is, $C' = \{x : \phi^*(x) = 0\}$. We now show that this process of obtaining C' from C is equivalent to applying discretistiaon of (5.10) to the initial curve C. The Euler-Lagrange equation for (5.11) reads

$$-d\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \left\langle \nabla d, \frac{\nabla \phi}{|\nabla \phi|} \right\rangle + \frac{1}{h} (\phi - s_C) = 0.$$

If we choose $x \in C'$, then $\phi^*(x) = 0$ and

$$\begin{aligned} s_C\,(x) &= -h \left[d\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \left\langle \nabla d, \frac{\nabla \phi}{|\nabla \phi|} \right\rangle \right](x) \\ &= -h \left[d\kappa - \langle \nabla d, \mathcal{N} \rangle \right](x), \end{aligned} \tag{5.12}$$

where κ and $\mathcal{N}$ respectively denote the mean curvature and normal to C'. If x_0 on C is the projection of x on C', and $\mathcal{N}(x_0)$ is the exterior normal to C. Then we have

$$x = x_0 + s_C(x) \mathcal{N}(x_0). \tag{5.13}$$

Then substituting (5.12) to (5.13) gives

$$x = x_0 - h \left[d\kappa - \langle \nabla d, \mathcal{N} \rangle \right] \mathcal{N}(x_0), \tag{5.14}$$

which corresponds to the discretisation of the GAC motion given in Equation (5.10), with time step h. Minimising solution of (5.11) thus provides an approximation of the evolution flow of GAC with time step h.

In summary, the following TV-based algorithm can be used for the evolution of the GAC model.

1: **while** "not converged" **do**
2: Define $\phi^{k+1} = \arg\min_\phi \{\int_\Omega d(x)\,|\nabla\phi| dx + \frac{1}{2h}\int_\Omega (\phi - s^k)^2 dx\}$
3: $s^{k+1} = SDF(\phi^{k+1})$
4: **end while**

To reconstruct a surface from the point cloud $\{x_i\}$, we need an additional data fidelity term related to the image $f(x)$ calculated from the point cloud. In this case, step 2 is modified to

$$\phi^{k+1} = \arg\min_\phi \left\{ \int_\Omega d(x)\,|\nabla\phi| dx + \frac{1}{2h}\int_\Omega \left(\phi - s^k\right)^2 dx + \lambda \int_\Omega Q(c_1, c_2)\phi dx \right\}. \tag{5.15}$$

In the algorithm, $SDF(\phi)$ denotes the signed distance function obtained by re-initialising ϕ, which can be achieved by applying the fast sweeping to the zero level set of ϕ, or using the reinitalisation method in [181–183]. The advantages of this algorithm are threefold: First, the TV-based formulation of the GAC model allows to use much larger time steps than the standard explicit discretisation, and thus the speed of the algorithm is not limited by the CFL condition [184]. Second, the model (5.11) is convex, meaning that given s_C there is an unique ϕ to be solved. This makes the algorithm robust to the initialisation of ϕ. Third, it can be easily and efficiently implemented using a fast TV solver, such as the split Bregman method introduced in Chapter 2 or [180]. However, the overall algorithm presented here is iterative, which means that several weighted TV problems need to be solved. Also, the signed distance function ϕ^k must be recomputed at each iteration. This makes it less efficient than the convex methods introduced in Sections 5.4.1 and 5.4.3.

5.4.3 TVG-L1

Another convexified binary image segmentation model is the TVG-L1 model proposed in [126]. which has the form of

$$\min_u \left\{ \int_\Omega g(x)\,|\nabla u| dx + \lambda \int_\Omega |u - f| dx \right\}, \tag{5.16}$$

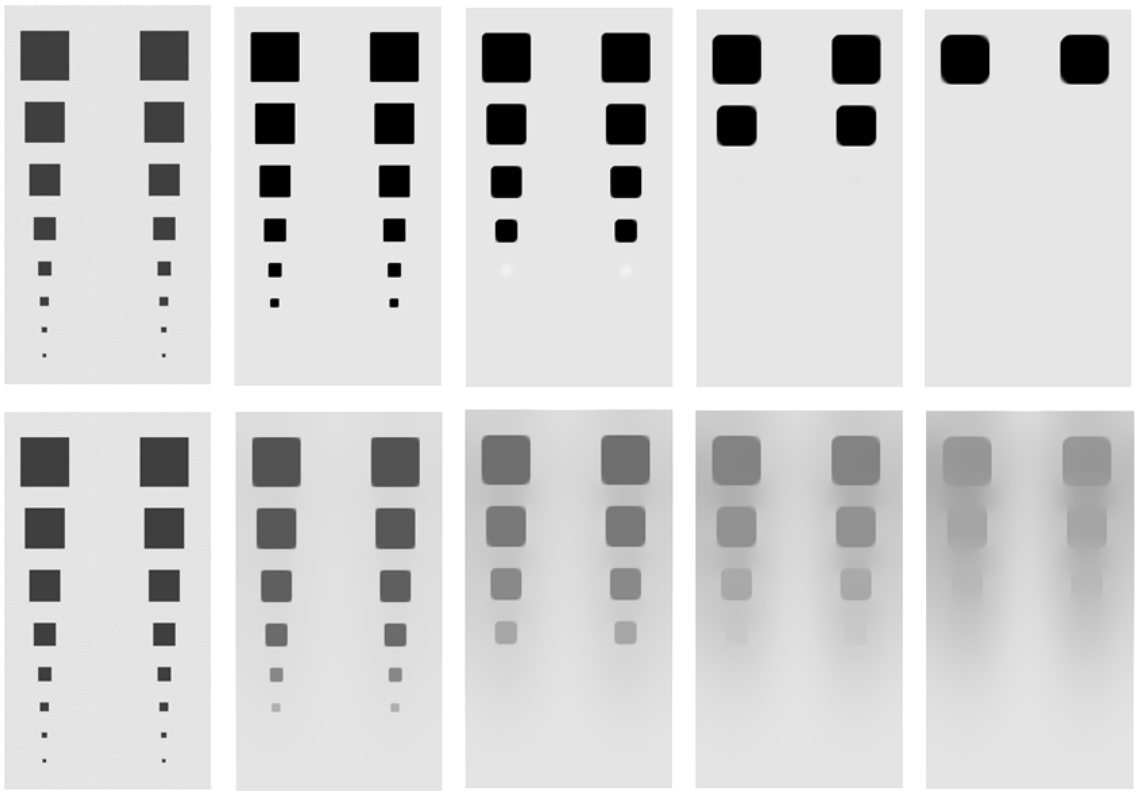


Figure 5.5: Comparison between the TVG-L1 model (5.16) and the TV model (5.17). 1st row: results from TGV-L1 with different λ. 2st row: results from TV with different λ. λ used in (5.16) and (5.17) is decreasing from left to right.

where the first term is the edge weighted TV-norm and the second term volumetric L1 data fidelity term. u is the function close to binary, and $f \in [0, 1]$ is the given image. Note that if we drop g in (5.16), the model is simply TV-L1. Next, let us recall the TV denoising minimisation problem and compare TV and TVG-L1 in detail

$$\min_u \left\{ \int_\Omega |\nabla u| dx + \lambda \int_\Omega (u - f)^2 dx \right\}. \tag{5.17}$$

The differences between them (5.17) and (5.16) are the introduction of the weighted TV-norm and the replacement of the L2-norm by the L1-norm as a fidelity measure. These modifications have two important consequences. First, the L1-norm, which has been widely studied in [185–188], outperforms the standard TV model with the L2-norm for some applications and presents important geometric properties concerning global minimisers of functionals. Second, the introduction of the weight function g in the TV-norm build a link between the GAC model and the weighted TV-norm. To further compare the standard TV model (5.17) with the TVG-L1 model (5.16), we develop the experiment shown in Figure 5.5.

As can be seen from Figure 5.5, TVG-L1 with the L1-norm fidelity better preserves the contrast and the order in which the squares disappear is completely determined in terms of the geometry (i.e., area) of the squares. Conversely, TV with the L2-norm fidelity results in contrast loss, does not have any geometry interpretation, and cannot preserve important geometric properties, such as corners.

We note that in the surface reconstruction problem, if f in TVG-L1 is close to a binary image, the minimiser is also close to a binary image with sharp transitions located at the places where $g(x)$ (i.e., $d(x)$) is small. Efficient algorithms are available for the convex TVG-L1 minimisation problem. For example, the split Bregman can be adopted. Here, we consider to use the first order primal dual projection method, which is fast and easier to implement. In the following, we develop such an algorithm for TVG-L1 for surface reconstruction.

First, we introduce an auxiliary variable v to decouple u in (5.16) and reformulate it to the problem

$$\min_{u,v} \left\{ \int_\Omega g|\nabla u| dx + \lambda \int_\Omega |v| dx + \frac{1}{2\theta} \int_\Omega (u+v-f)^2 dx \right\}, \tag{5.18}$$

where θ is a positive penalty parameter. In theory, if $\theta \to 0$, (5.18) is exactly the TVG-L1 model (5.16). However, practice showed that the algorithm is robust even when θ is large. The problem (5.18) is a convex optimisation problem with respect to two variables u and v, which can be solved efficiently with the fast alternating minimisation technique. We first solve the problem (5.18) for u by fixing v. The resulting minimisation problem becomes

$$\min_{u} \left\{ \int_\Omega g|\nabla u| dx + \frac{1}{2\theta} \int_\Omega (u+v-f)^2 dx \right\}, \tag{5.19}$$

which is similar to the TV model (5.17). The only difference lies in the edge weighted TV-norm used in (5.19). We can solve this problem using the fast primal dual projection method proposed in [189] with an extrapolation step as follows

$$\begin{cases} \xi^{k+1} = \mathrm{Proj}_K \left(\xi^k - \sigma \nabla U^k \right) \\ u^{k+1} = f - v - \theta \nabla \cdot \xi^{k+1} \\ U^{k+1} = u^{k+1} + \alpha \left(u^{k+1} - u^k \right) \end{cases}, \tag{5.20}$$

where ξ is the dual variable and σ is a time step. In order to ensure that the iterative scheme remains stable, $\sigma \leq 1/8$. $\alpha \in (0,2)$ is an over- or under-relaxation factor. It is normally set to 1 for fast convergence. Proj_K denotes the back projection on set K, i.e.,

$\{x : |\xi(x)| \leq 1\}$, and it is defined as

$$\mathrm{Proj}_K(\xi) = \frac{\xi}{\max(|\xi|, g)}.$$

After u is solved using (5.20), we need to optimise the following problem for v

$$\min_v \left\{ \lambda \int_\Omega |v| dx + \frac{1}{2\theta} \int_\Omega (u + v - f)^2 dx \right\}, \tag{5.21}$$

which is solved analytically by the 1D shrinkage equation as follows

$$v = \begin{cases} f - u - \lambda\theta & \text{if } f - u \geq \lambda\theta \\ f - u + \lambda\theta & \text{if } f - u \leq -\lambda\theta \\ 0 & \text{if } |f - u| \leq \lambda\theta \end{cases}.$$

5.5 The proposed variational level set method

We propose a local optimisation technique based on the level set method for the surface reconstruction task. There are two main reasons for this choice: 1) in numerical studies, our reconstructed targets are all single closed object, so the proposed model has few minima such that local optimisers can be expected to produce similar results as those of global optimisers; 2) compared to the aforementioned convex optimisation problems, in our case even local extrema represent a sensible segmentation of a given input image. In fact, in our case a local optimum is often more desirable, because it corresponds to the closest segmentation for a given initialisation. The truth is that we have already obtained a nice initialisation that is very close to true segmentation in Section 5.2.2.

5.5.1 The proposed variational model

The functional of the proposed variational model consists of three energy terms, which read

$$E(\phi, c_1, c_2) = E_R(\phi) + \lambda E_I(\phi, c_1, c_2) + \beta E_B(\phi). \tag{5.22}$$

The first term E_R is a regularisation term to keep the reconstructed surface smooth while maintain the level set function ϕ as a signed distance function:

$$E_R(\phi) = \int_\Omega d(x) |\nabla H(\phi)| dx + \frac{\mu}{2} \int_\Omega (|\nabla\phi| - 1)^2 dx, \tag{5.23}$$

where $d(x)$ is the distance function calculated from the original point cloud $\{x_i\}$ in Section 5.2.1, and μ is a positive parameter penalising the deviation of ϕ from a signed distance function. Larger μ leads to more similarity between ϕ and the signed distance function. The first term in this functional is equivalent to the GAC model (5.1) under the level set framework. It still can be understood as the weighted TV-norm as before. The second term is to ensure that the level set function ϕ remains a signed distance function, thus eliminating the need of reinitialisation.

The second data fitting term E_I incorporates the information derived from the dataset which is given by

$$E_I(c_1, c_2, \phi) = \int_\Omega Q(c_1, c_2) H(\phi) dx, \tag{5.24}$$

where c_1 and c_2 represent the mean values inside and outside of the zero level set of ϕ. $Q(c_1, c_2) = (c_1 - f(x))^2 - (c_2 - f(x))^2$, and $f(x)$ is the image computed from Section 5.2.2. This term follows the work of the two-phase piecewise constant Chan-Vese model [177] to maintain regions of similar intensity values. The term can help improve the accuracy of reconstruction by capturing fine features of the object with similar intensity values. The penalty parameter λ on this term in (5.22) is positive.

The third balloon force term E_B includes the area information inside the zero level set of ϕ to speed up surface evolution as well as segment concave objects:

$$E_B(\phi) = \int_\Omega d(x) H(-\phi) dx. \tag{5.25}$$

The parameter β for this term can be positive or negative depending on whether inside or outside of the zero level set. If the initial boundary is placed outside the object, the coefficient takes a positive values so that the zero level set can shrink during level set evolution. If the initial boundary is placed inside the object, the coefficient takes a negative value to expand the boundary.

The proposed model (5.22) can be solved by an optimization procedure. First ϕ is fixed to optimise c_1 and c_2 as follows

$$c_1 = \frac{\int_\Omega f H(\phi) dx}{\int_\Omega H(\phi) dx} \quad \text{and} \quad c_2 = \frac{\int_\Omega f(1 - H(\phi)) dx}{\int_\Omega (1 - H(\phi)) dx}.$$

Then c_1 and c_2 are fixed using the following gradient descent flow starting with $\phi = \phi_0$ to minimise (5.22), where ϕ_0 comes from Section 5.2.2.

$$\frac{\partial \phi}{\partial t} = \left(\nabla \cdot \left(d \frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda Q(c_1, c_2) + \beta d \right) \delta(\phi) + \mu \left(\Delta \phi - \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right). \tag{5.26}$$

In practice, the Heaviside function $H(\phi)$ and Dirac function $\delta(\phi)$ in (5.22) and (5.26) are approximated by their regularised versions with a small positive number ϵ

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

$$\delta_\epsilon(\phi) = \frac{1}{\pi}\frac{\epsilon}{\epsilon^2 + \phi^2}.$$

5.5.2 3D discretisation with finite difference scheme

$$\begin{aligned}
\nabla \cdot \left(d\frac{\nabla\phi}{|\nabla\phi|_\varepsilon}\right)_{i,j,k} = {} & d_{i,j+\frac{1}{2},k}\frac{\partial_x^+\phi_{i,j,k}}{\sqrt{\left(\nabla_x^+\phi_{i,j,k}\right)^2 + \left(\nabla_y^0\phi_{i,j+\frac{1}{2},k}\right)^2 + \left(\nabla_z^0\phi_{i,j+\frac{1}{2},k}\right)^2 + \varepsilon^2}} \\
& - d_{i,j-\frac{1}{2},k}\frac{\partial_x^-\phi_{i,j,k}}{\sqrt{\left(\nabla_x^-\phi_{i,j,k}\right)^2 + \left(\nabla_y^0\phi_{i,j-\frac{1}{2},k}\right)^2 + \left(\nabla_z^0\phi_{i,j-\frac{1}{2},k}\right)^2 + \varepsilon^2}} \\
& + d_{i+\frac{1}{2},j,k}\frac{\partial_y^+\phi_{i,j,k}}{\sqrt{\left(\nabla_y^+\phi_{i,j,k}\right)^2 + \left(\nabla_x^0\phi_{i+\frac{1}{2},j,k}\right)^2 + \left(\nabla_z^0\phi_{i+\frac{1}{2},j,k}\right)^2 + \varepsilon^2}} \\
& - d_{i-\frac{1}{2},j,k}\frac{\partial_y^-\phi_{i,j,k}}{\sqrt{\left(\nabla_y^-\phi_{i,j,k}\right)^2 + \left(\nabla_x^0\phi_{i-\frac{1}{2},j,k}\right)^2 + \left(\nabla_z^0\phi_{i-\frac{1}{2},j,k}\right)^2 + \varepsilon^2}} \\
& + d_{i,j,k+\frac{1}{2}}\frac{\partial_z^+\phi_{i,j,k}}{\sqrt{\left(\nabla_z^+\phi_{i,j,k}\right)^2 + \left(\nabla_x^0\phi_{i,j,k+\frac{1}{2}}\right)^2 + \left(\nabla_y^0\phi_{i,j,k+\frac{1}{2}}\right)^2 + \varepsilon^2}} \\
& - d_{i,j,k-\frac{1}{2}}\frac{\partial_z^-\phi_{i,j,k}}{\sqrt{\left(\nabla_z^-\phi_{i,j,k}\right)^2 + \left(\nabla_x^0\phi_{i,j,k-\frac{1}{2}}\right)^2 + \left(\nabla_y^0\phi_{i,j,k-\frac{1}{2}}\right)^2 + \varepsilon^2}}
\end{aligned} \tag{5.27}$$

In order to evolve the level set function ϕ in (5.26), discretisations are needed for $\nabla \cdot (d\frac{\nabla\phi}{|\nabla\phi|})$ and $\nabla \cdot (\frac{\nabla\phi}{|\nabla\phi|})$ in 3D based on the finite difference scheme. Let $\Omega \to \mathbb{R}^{MNL}$ denote the 3D grid space of size MNL. The second order coupled (with the distance function d) curvature term $\nabla \cdot (d\frac{\nabla\phi}{|\nabla\phi|})$ at voxel (i, j, k) can be discretised as (5.27), where ϵ is a small positive number to avoid division by zero. Note that the half-point difference scheme is used here for (5.27) in order to satisfy rotation-invariant characteristics. The distance function d on half-points between each two integer voxels are given as

$$d_{i,j+\frac{1}{2},k} = \frac{d_{i,j+1,k} + d_{i,j,k}}{2}, \quad d_{i,j-\frac{1}{2},k} = \frac{d_{i,j-1,k} + d_{i,j,k}}{2}$$

$$d_{i+\frac{1}{2},j,k} = \frac{d_{i+1,j,k} + d_{i,j,k}}{2}, \quad d_{i-\frac{1}{2},j,k} = \frac{d_{i-1,j,k} + d_{i,j,k}}{2}$$

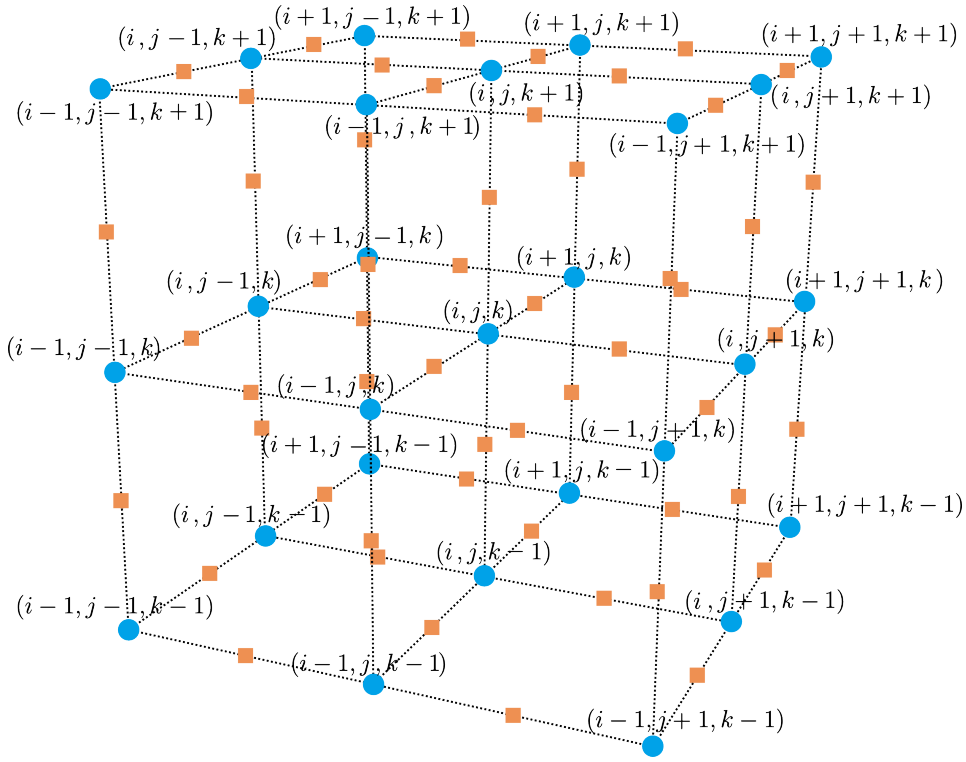


Figure 5.6: 3D grid space for calculating the discrete differential operators used in equation (5.27). Sphere dots (yellow) represent voxels. Cube points (blue) are half points between two integer voxels.

$$d_{i,j,k+\frac{1}{2}} = \frac{d_{i,j,k+1} + d_{i,j,k}}{2}, \quad d_{i,j,k-\frac{1}{2}} = \frac{d_{i,j,k-1} + d_{i,j,k}}{2}$$

The first order forward ∂_x^+ and backward ∂_x^- discrete derivatives along x, y and z directions on voxel (i, j, k) can be defined as

$$\partial_x^+ \phi_{i,j,k} = \phi_{i,j+1,k} - \phi_{i,j,k}, \quad \partial_x^- \phi_{i,j,k} = \phi_{i,j,k} - \phi_{i,j-1,k}$$
$$\partial_y^+ \phi_{i,j,k} = \phi_{i+1,j,k} - \phi_{i,j,k}, \quad \partial_y^- \phi_{i,j,k} = \phi_{i,j,k} - \phi_{i-1,j,k}$$
$$\partial_z^+ \phi_{i,j,k} = \phi_{i,j,k+1} - \phi_{i,j,k}, \quad \partial_z^- \phi_{i,j,k} = \phi_{i,j,k} - \phi_{i,j,k-1}$$

The central differences are applied to approximate the following discrete first order derivatives on half-points between each two integer voxels in (5.27).

$$\nabla_y^0 \phi_{i,j+\frac{1}{2},k} = \frac{\phi_{i+1,j+1,k} + \phi_{i+1,j,k} - \phi_{i-1,j+1,k} - \phi_{i-1,j,k}}{4}$$
$$\nabla_z^0 \phi_{i,j+\frac{1}{2},k} = \frac{\phi_{i,j+1,k+1} + \phi_{i,j,k+1} - \phi_{i,j+1,k-1} - \phi_{i,j,k-1}}{4}$$

$$\nabla_y^0 \phi_{i,j-\frac{1}{2},k} = \frac{\phi_{i+1,j,k} + \phi_{i+1,j-1,k} - \phi_{i-1,j,k} - \phi_{i-1,j-1,k}}{4}$$

$$\nabla_z^0 \phi_{i,j-\frac{1}{2},k} = \frac{\phi_{i,j,k+1} + \phi_{i,j-1,k+1} - \phi_{i,j,k-1} - \phi_{i,j-1,k-1}}{4}$$

$$\nabla_x^0 \phi_{i+\frac{1}{2},j,k} = \frac{\phi_{i+1,j+1,k} + \phi_{i,j+1,k} - \phi_{i+1,j-1,k} - \phi_{i,j-1,k}}{4}$$

$$\nabla_z^0 \phi_{i+\frac{1}{2},j,k} = \frac{\phi_{i+1,j,k+1} + \phi_{i,j,k+1} - \phi_{i+1,j,k-1} - \phi_{i,j,k-1}}{4}$$

$$\nabla_x^0 \phi_{i-\frac{1}{2},j,k} = \frac{\phi_{i,j+1,k} + \phi_{i-1,j+1,k} - \phi_{i,j-1,k} - \phi_{i-1,j-1,k}}{4}$$

$$\nabla_z^0 \phi_{i-\frac{1}{2},j,k} = \frac{\phi_{i,j,k+1} + \phi_{i-1,j,k+1} - \phi_{i,j,k-1} - \phi_{i-1,j,k-1}}{4}$$

$$\nabla_x^0 \phi_{i,j,k+\frac{1}{2}} = \frac{\phi_{i,j+1,k+1} + \phi_{i,j+1,k} - \phi_{i,j-1,k+1} - \phi_{i,j-1,k}}{4}$$

$$\nabla_y^0 \phi_{i,j,k+\frac{1}{2}} = \frac{\phi_{i+1,j,k+1} + \phi_{i+1,j,k} - \phi_{i-1,j,k+1} - \phi_{i-1,j,k}}{4}$$

$$\nabla_x^0 \phi_{i,j,k-\frac{1}{2}} = \frac{\phi_{i,j+1,k} + \phi_{i,j+1,k-1} - \phi_{i,j-1,k} - \phi_{i,j-1,k-1}}{4}$$

$$\nabla_y^0 \phi_{i,j,k-\frac{1}{2}} = \frac{\phi_{i+1,j,k} + \phi_{i+1,j,k-1} - \phi_{i-1,j,k} - \phi_{i-1,j,k-1}}{4}$$

To discretise the curvature term $\nabla \cdot (\frac{\nabla \phi}{|\nabla \phi|})$, we set $d_{i,j+\frac{1}{2},k} = d_{i,j-\frac{1}{2},k} = d_{i+\frac{1}{2},j,k} = d_{i-\frac{1}{2},j,k} = d_{i,j,k+\frac{1}{2}} = d_{i,j,k-\frac{1}{2}} = 1$ in (5.27).

5.6 Experiments

Some 2D and 3D reconstruction results using the proposed method are presented. In Figure 5.7, a 2D contour is given of the dataset shown in Figure 5.1 (a). The initial contour obtained in Section 5.2.2 is very close to the true point cloud, which can speed up convergent rate. As the evolution proceeds, the reconstruction by Zhao's method [2] loses the small features of the original data (i.e., the two convex parts). However, the term (5.24) in the proposed model is able to preserve these features.

Figure 5.8 shows the reconstructed 3D surface of the Bunny point cloud shown in Figure 5.1 (b). Zhao's method failed to reconstruct Bunny's ears and feet, while the proposed method succeeded. In addition, we compare our method with the Poisson reconstruction [172], where the authors show that surface reconstruction from oriented points can be cast as a spatial Poisson problem. They considered the relationship between the gradient of indicator function and an integral of the surface normal field. In detail,

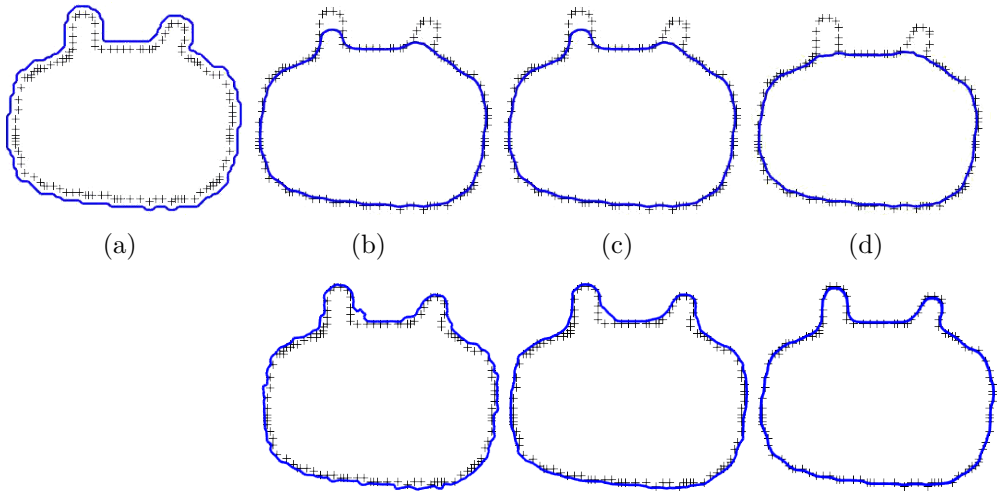


Figure 5.7: Comparison of the proposed model (5.22) with the Zhao's method described in [2]. (a): Initialisation for both methods; (b)-(d): Intermediate and final results by Zhao's method; (e)-(g): Intermediate and final results by our method with $\beta = 0$.

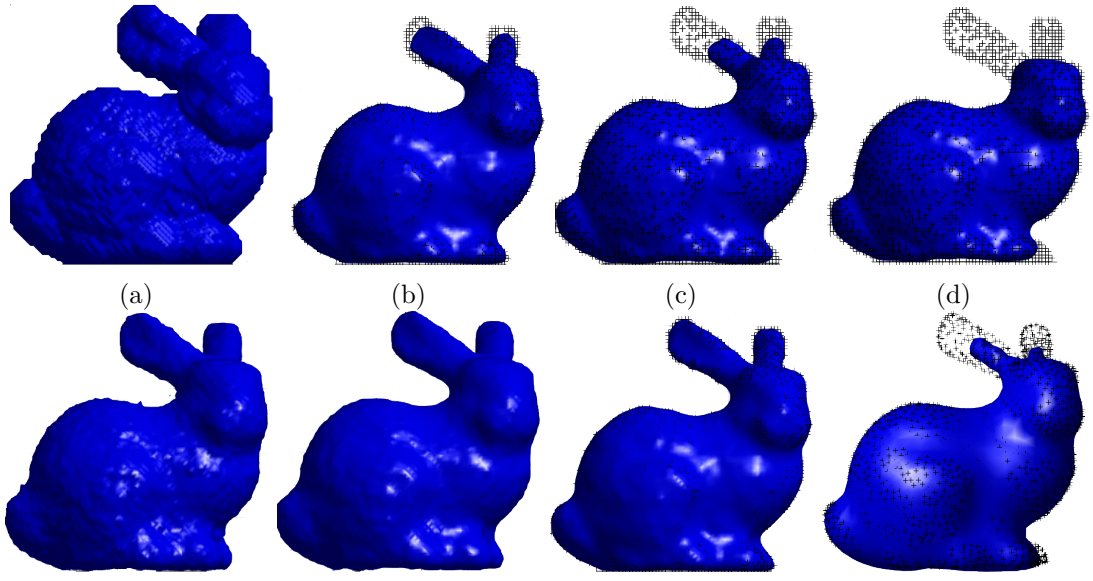


Figure 5.8: Comparison with state-of-the-art. (a): Same initialisation for models to be compared; (b)-(d): Intermediate and final results by Zhao's method; (e)-(g): Intermediate and final results by our method with $\beta = 0$; (h): Result by Poisson.

the gradient of an indicator function should be zero nearly everywhere except these points on a surface where their gradients are consistent with the inward surface normals. Based on this, the Poisson reconstruction problem can be simply deemed as calculating indicator function from the gradient field of a surface. As can be seen in this experiment,

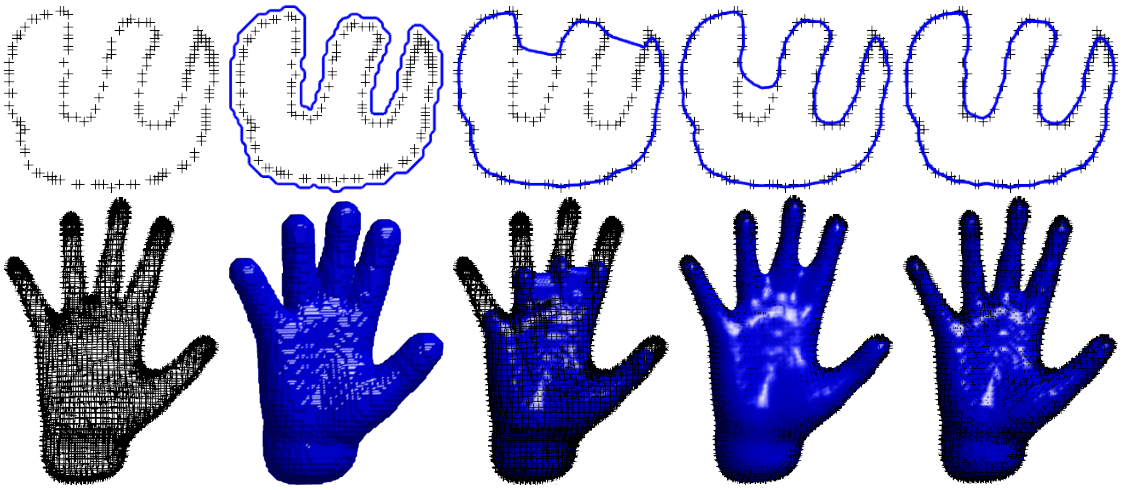
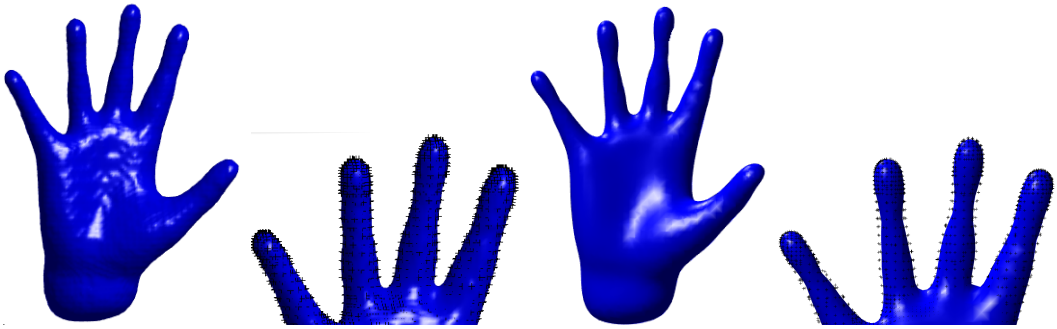


Figure 5.9: Effectiveness of the balloon force term (5.25) in the proposed model using 2D and 3D point clouds. 1st column: original data points; 2nd column: initialisation obtained using the method proposed in Section 5.2.2; 3rd column: results by Zhao's method; 4th column: results by the proposed model without using balloon force term (i.e., $\beta = 0$ in (5.22)); 5th column: results by the proposed model.



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

Figure 5.10: Comparison with Poisson. (a) and (b) are reconstructions by the proposed model; (c) and (d) are reconstructions by the Poisson method; (b) and (d) are zoomed-in versions of (a) and (c), with the original data points added.

the reconstruction of Bunny by Poisson loses some texture and looks smoother than that by the proposed method. This experiment demonstrates the effectiveness of the image data fitting term (5.24) in the proposed model.

Figure 5.9 shows the reconstruction results of a 2D concave object and a 3D human hand that contains fine details (i.e., fingers) and concave regions (i.e., the spaces between fingers). Zhao's method gets stuck in the concave regions and also loses the fingers. The proposed model with only the (5.24) term can preserve the fingers and partially go down the concave regions. The proposed model with both terms (5.24) and (5.25) succeeds

in preserving all features as well as the concave regions in both 2D and 3D cases. This validates the capability of the balloon force term (5.25) in the proposed model.

Figure 5.11: Surface reconstruction from point clouds using the proposed method. 1st and 3rd rows: point clouds; 2nd and 4th rows: reconstructed results.

Figure 5.10 shows that Poisson result is smoother than that of the proposed method, but some fine details/texture, i.e., palm prints, are smeared by the Poisson method.

Figure 5.10 (b) and (d) show that the Poisson result also misses several original data points, and the reconstructed fingers by Poisson are thinner than those by the proposed method. The proposed method thus performs better than the Poisson method.

Figure 5.11 shows the surface reconstruction results from the point clouds that contain more complex geometries. As can be seen, the proposed method performs consistently well for these complicated objects. The reconstructed surfaces are smooth and visually pleasant due to the continuous nature of the signed distance function used in our method. We also implemented the binary based segmentation models, such as the convex method (5.8) or the TVG-L1 model (5.16), and found that the resulting reconstructed shapes are not as smooth as the proposed method. This is because the binary representation is discontinuous as compared to the continuous level set function in our method.

5.7 Summary

In this chapter, a novel variational level set method is proposed to reconstruct implicit surfaces from unorganised point clouds. Implementation details of the variational model are given and the three energy functional terms are illustrated through numerical experiments. Major advantages of the proposed method over existing approaches include accuracy and detail preservation, without the need for reinitialisation. Since the proposed method effectively integrates the region and edge information, it would not get stuck at the local minimum which is commonly associated with edge-based implicit reconstruction methods. The components of the proposed variational model are explained and details of the discretisation procedure for implementing the variational model are also given.

Chapter 6

Conclusion and future work

In this thesis, we have systematically studied modern variational and PDE-based methods and their applications to image processing, such as image denoising, image segmentation and surface reconstruction. While image denoising is used as a pre-processing tool, the thesis is more focusing on image segmentation problems. For example, we have successfully applied these approaches to detect and quantify malaria parasites from red blood cell (RBC) images, extract micgrila cells from histology images, delineate retinal layer boundaries from optical coherence tomography (OCT) images, and reconstruct surfaces from a set of unorganised point clouds. Technically, we have introduced how these approaches can be numerically implemented in detail, and validated the performance and effectiveness of our methods via extensive experimental results as well as comparison with state-of-the-art methods

6.1 Contributions

This thesis consists of four main distinguishable parts associated with the publications that have been produced by the author as a part of his PhD [24, 94, 92, 93, 161]. In what follows, we conclude our contributions which have been introduced in more details in Chapters 2, 3, 4 and 5.

- We performed a systematic comparison of up to 7 (6 higher order) variational methods for denoising images with different features. The comparison includes abilities of sharp edges recovery, smoothness preservation, robustness against noise levels, as well as computational efficiency. We discretised various first, second, fourth order partial derivatives using the finite difference method and solved the compared variational models under a unified framework—the FFT-based split Bergman. To

our knowledge, this is the first systematic study in the area of variational image denoising in terms of so comprehensive numerical implementations and results provided.

- We developed an image analysis framework for an image-based cytometer, which detects and numerically counts malaria parasite infected red blood cells (iRBCs) from Giemsa-stained smears derived from iRBCs. Our framework is able to accurately classify all parasitic subpopulations by quantifying the area occupied by the parasites within iRBCs. Moreover, we compared the efficacy of our proposed image algorithms (used in the image-based cytometer) against a commercial flow cytometer and demonstrate comparable results between the two methods. Collectively, these results highlighted the possibility to use our image analysis framework together with the image-based cytometer as a cheap, rapid and accurate alternative for antimalarial testing without compromising on efficiency and minimal processing time. Our image analysis framework therefore might be used for field diagnosis of malaria.

- We developed a method for neuroscientists to quantitatively analyse microglia histology data. Microglial cells are segmented using the Mumford-Shah total variation, fast split Bregman method, and adaptive thresholding. We show that this framework offers advantages over existing image analysis methods by validating against manual analysis of imaging data from wild type mice and a transgenic mouse model of Alzheimer's disease. The proposed segmentation method is fast, and accurate and robust. These properties ensure that the proposed segmentation method is scalable to large datasets, allowing the analysis of microglia in regions of interest as well as across the whole brain.

- We proposed a fully automated method that is able to segment healthy and pathological retinal layers from optical coherence tomography images. The method uses a weighted geodesic distance which is efficiently calculated from an Eikonal equation via the fast sweeping method. Segmentation then proceeds by solving an ordinary differential equation for the geodesic distance. The contribution of this work is twofold, clinically and technically: 1) providing a precise individual OCT segmentation system for clinical use; 2) introducing an OCT-specific weight function into the geodesic distance framework. To our knowledge, this is the first work on 2D/3D segmentation of intra-retinal layer structures using the geodesic distance method.

- We viewed the problem of surface reconstruction from point cloud as the variational image segmentation problem, and proposed a novel variational level set method for surface reconstruction. The proposed variational model is capable of accurately reconstructing smooth surfaces from point clouds whilst preserving the fine details of the underlying point cloud. The model also completely eliminated the need for reinitialisation, a common time-consuming procedure associated with the level set method. Extensive experiments are conducted to show that the proposed method outperformed the state-of-the-art surface reconstruction approaches

6.2 Future work

The image segmentation methods proposed in this thesis have achieved significant success in automatically segmenting RBC, histology and OCT images, but there is still room for improvement of these methods. For example, the above automatic segmentation approaches only considered gray intensity values or image gradient as a feature. As a result, they tend to fail to segment meaningful objects from images when the objects are occluded by other objects or some parts of them are in low signal to noise or even missing. These situations in fact always exist in a lot of medical imaging applications. In these cases, shape information can be used to successfully extract the desirable objects from low quality images. Studying segmentation methods incorporated with shape prior is therefore one of our future work.

Image segmentation in essence is about how to extract good features that saliently represent the objects of interest in an image. While image intensity and gradient are two conventional options sometimes insufficient to represent complex segmentation targets, good feature extractors designed by hand normally require a considerable amount of engineering skill and domain expertise. This however can be avoided if good features can be learned automatically using a general-purpose learning procedure. This is why deep learning starts to play an important role in medical image analysis by automatically learning morphological and/or textural patterns from images without using man craft features. Deep learning methods have achieved state-of-the-art performance across various medical image segmentation applications. Breakthrough improvements were achieved by using a large data set of medical images from which deep models can find more generalised features and thus lead to improved performance. Our next future work therefore will focus on image segmentation using deep learning techniques.

Appendix A

Fast Fourier transform solver

This Appendix details how to implement the fast Fourier transform to solve a second order linear PDE. Normally, if one uses the variable splitting algorithm for a total variation type optimisation problem, one of the resulting equations is the linear elliptic PDE, which has the form of

$$u(x) - \theta \Delta u(x) = g(x), \tag{A.1}$$

with a suitable boundary condition often imposed by a specific problem. A numerical solution for (A.1) on a rectangle grid can be sought via the finite difference method. If we rewrite (A.1) using the discrete definition of the Laplace differential operator, we have

$$u_{i,j} - \theta \left(u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j} - 4u_{i,j} \right) = g_{i,j}. \tag{A.2}$$

Now let us transform $u_{i,j}$ into frequency domain via the discrete Fourier transform as follows

$$\mathcal{F}\left(u_{i,j}\right) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} u_{i,j} e^{-\sqrt{-1}\left(\frac{2\pi ri}{M} + \frac{2\pi sj}{N}\right)} \tag{A.3}$$

where $\mathcal{F}$ denotes the discrete Fourier transform. $\mathcal{F}\left(u_{i,j}\right)$ is the map of $u_{i,j}$ in the Fourier domain. i, j are the discrete indices in the discrete time domain, while r, s are the discrete indices in the frequency domain. $i \in [1, M]$ and $j \in [1, N]$, $r \in [0, M)$ and $s \in [0, N)$. For the discrete frequencies, r and s, we have

$$\begin{aligned} \mathcal{F}\left[u_{i\pm1,j}\right] &= e^{\pm\sqrt{-1}\frac{2\pi r}{M}} \mathcal{F}\left(u_{i,j}\right), \\ \mathcal{F}\left[u_{i,j\pm1}\right] &= e^{\pm\sqrt{-1}\frac{2\pi s}{N}} \mathcal{F}\left(u_{i,j}\right). \end{aligned} \tag{A.4}$$

(A.4) is known as the time shift property of the discrete Fourier transform. In addition, note that (A.4) holds only if the periodical boundary condition on u is imposed, which means that if the FFT is applied to solve the linear PDE, the periodical boundary condition is automatically assumed.

Applying the discrete Fourier transform to the both sides of the equation (A.2), we have

$$\mathcal{F}\left[u_{i,j} - \theta\left(u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j} - 4u_{i,j}\right)\right] = \mathcal{F}\left[g_{i,j}\right]. \tag{A.5}$$

In (A.5), let

$$U_{i,j} = u_{i,j} - \theta\left(u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j} - 4u_{i,j}\right),$$

and with (A.4) we have the following formulation for the left-hand side of (A.5)

$$\mathcal{F}\left[U_{i,j}\right] = \left[1 - \theta\left(e^{\sqrt{-1}\frac{2\pi s}{N}} + e^{-\sqrt{-1}\frac{2\pi s}{N}} + e^{\sqrt{-1}\frac{2\pi r}{M}} + e^{-\sqrt{-1}\frac{2\pi r}{M}} - 4\right)\right]\mathcal{F}\left(u_{i,j}\right). \tag{A.6}$$

With the help of the famous Euler equation $e^{\pm\sqrt{-1}x} = \cos x \pm \sqrt{-1}\sin x$, we can obtain an equivalent form of (A.6) in terms of the trigonometric function form, which is

$$\mathcal{F}\left[U_{i,j}\right] = \left[1 - \theta\begin{pmatrix} \cos\frac{2\pi s}{N} + \sqrt{-1}\sin\frac{2\pi s}{N} \\ + \cos\frac{2\pi s}{N} - \sqrt{-1}\sin\frac{2\pi s}{N} \\ + \cos\frac{2\pi r}{M} + \sqrt{-1}\sin\frac{2\pi r}{M} \\ + \cos\frac{2\pi r}{M} - \sqrt{-1}\sin\frac{2\pi r}{M} - 4 \end{pmatrix}\right]\mathcal{F}\left(u_{i,j}\right),$$

which is equal to

$$\mathcal{F}\left[U_{i,j}\right] = \underbrace{\left[1 - 2\theta\left(\cos\frac{2\pi r}{M} + \cos\frac{2\pi s}{N} - 2\right)\right]}_{\kappa}\mathcal{F}\left(u_{i,j}\right) = \mathcal{F}\left[g_{i,j}\right],$$

which leads to a closed-form solution of u as

$$u_{i,j} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}\left[g_{i,j}\right]}{\kappa}\right)\right), \tag{A.7}$$

where $\mathcal{F}^{-1}$ denotes the discrete inverse Fourier transform. $\Re(\cdot)$ is the real part of a complex number. "—" stands for pointwise division of matrices.

Appendix B

Fast sweeping

In both OCT segmentation and surface reconstruction problems, we intend to solve a distance function from a first order nonlinear PDE. More specifically, let us consider the following generalised Eikonal equation

$$|\nabla d(x)| = f(x),\ \forall x \in (\Omega \backslash \Gamma) \subset \mathbb{R}^m \tag{B.1}$$

with

$$d(x) = 0,\ \forall x \in \Gamma \subset \mathbb{R}^m. \tag{B.2}$$

Above $m = 2$ and $m = 3$ respectively correspond to 2D and 3D cases. For OCT segmentation, $\Gamma = \{s_2\}$ where $\{s_2\}$ is a seed point in 2D or multiple seed points in 3D, and $f(x)$ is $W^{-1}(x)$ where the weight function W is defined in (4.4). For surface reconstruction, $\Gamma = \{x_i\}$ where $\{x_i\}$ represent the positions of the scattered points, and $f(x)$ is 1 or the annular binary input. For 2D implementation, we use $x_{i,j}$ to denote a pixel point in the computational domain Ω, and $d_{i,j}$ to denote the numerical solution at $x_{i,j}$. For 3D implementation, we use $x_{i,j,k}$ to denote a voxel point in Ω, and $d_{i,j,k}$ to denote the numerical solution at $x_{i,j,k}$. The pixel or voxel size is set to 1 for all the cases.

B.1 2D Implementation

The 2D Godunov upwind difference scheme is used to discretise (B.1) as follows

$$\left[(d^n_{i,j} - d^n_{xmin})^+\right]^2 + \left[(d^n_{i,j} - d^n_{ymin})^+\right]^2 = f^2_{i,j,k}, \tag{B.3}$$

where $d^n_{xmin} = min(d^n_{i,j+1}, d^n_{i,j-1})$, $d^n_{ymin} = min(d^n_{i+1,j}, d^n_{i-1,j})$ and

$$x^+ = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}. \tag{B.4}$$

Boundary conditions need to be handled appropriately for (B.3) in the computational grid space Ω. A one-sided upwind difference is used for each of 4 boundaries in the grid space. For example, at the left boundary $x_{i,1}$, a one-sided difference along the x direction is computed as

$$\left[(d^n_{i,1} - d^n_{i,2})^+\right]^2 + \left[(d^n_{i,1} - d^n_{ymin})^+\right]^2 = f^2_{i,1}.$$

If we denote $a = d^n_{xmin}$ and $b = d^n_{ymin}$, numerically solving the equation (B.3) with its boundary condition needs to iteratively update

$$d_{i,j} = min(d^n_{i,j}, \widetilde{d_{i,j}}), \tag{B.5}$$

where

$$\widetilde{d^{n+1}_{i,j}} = \begin{cases} \min(a,b) + f_{i,j} & |a-b| \geq f_{i,j} \\ \frac{a+b\sqrt{2f^2_{i,j} - (a-b)^2}}{2} & |a-b| < f_{i,j} \end{cases}. \tag{B.6}$$

Initialisation. The hard constraint $d(x) = 0, x \in \Gamma$ in (B.2) should be satisfied over iterations. For initialisation, we assign exact values $d(x) = 0$ for the pixel points on Γ. These values are forced to be zeros at each iteration to satisfy such hard constraint. For the rest of pixel points in Ω, we assign large positive values for $d(x)$. These values will be updated later.

Gauss-Seidel iterations with alternating sweeping orderings. (B.5) is not analytical so iterations are needed to seek its numerical solution. The Gauss-Seidel iterative method is used here for fast convergence. There are different sweeping schemes that can be applied to Gauss-Seidel, such as the red-black sweeping, Lexicographic ordering sweeping, etc. In [175], the whole domain Ω is swept with the four alternating orderings repeatedly

$$(1)\ i = 1:M, j = 1:N;\ (2)\ i = M:1, j = N:1;$$
$$(3)\ i = 1:M, j = N:1;\ (4)\ i = M:1, j = 1:N.$$

Note that for simple geometry of Γ it may be sufficient for (B.5) to converge after applying such sweeping technique only once. However, for non-uniform problems and/or complex

geometry, such sweeping might be required to repeat multiple times in order for (B.5) to converge.

B.2 3D Implementation

The 3D Godunov upwind difference scheme is used to discretise (B.1) as follows

$$\left[(d^n_{i,j,k} - d^n_{xmin})^+\right]^2 + \left[(d^n_{i,j,k} - d^n_{ymin})^+\right]^2 + \left[(d^n_{i,j,k} - d^n_{zmin})^+\right]^2 = f^2_{i,j,k}. \tag{B.7}$$

In Equation (B.7), $d^n_{xmin} = min(d^n_{i,j+1,k}, d^n_{i,j-1,k})$, $d^n_{ymin} = min(d^n_{i+1,j,k}, d^n_{i-1,j,k})$, $d^n_{zmin} = min(d^n_{i,j,k+1}, d^n_{i,j,k-1})$ and x^+ is defined as (B.4). Boundary conditions need to be handled as well in the grid space Ω. The one-sided upwind difference is used for each of the 6 boundary faces of the grid space. For example, at the left boundary face, a one-sided difference along the x direction is computed as

$$\left[(d^n_{i,1,k} - d^n_{i,2,k})^+\right]^2 + \left[(d^n_{i,1,k} - d^n_{ymin})^+\right]^2 + \left[(d^n_{i,1,k} - d^n_{zmin})^+\right]^2 = f^2_{i,1,k}.$$

d^n_{xmin}, d^n_{ymin} and d^n_{zmin} are then sorted in a increasing order and the sorted version is recorded as a_1, a_2 and a_3. So, the unique solution to (B.7) is given as follows:

$$d^{n+1}_{i,j,k} = min(d^n_{i,j,k}, \widetilde{d_{i,j,k}}), \tag{B.8}$$

where $\widetilde{d_{i,j,k}}$ is a piecewise function containing three parts

$$\widetilde{d_{i,j,k}} = \begin{cases} \frac{1}{3}\left(a_1 + a_2 + a_3 + \sqrt{3f^2_{i,j,k} - (a_1 - a_2)^2 - (a_1 - a_3)^2 - (a_2 - a_3)^2}\right) \\ \frac{1}{2}\left(a_1 + a_2 + \sqrt{2f^2_{i,j,k} - (a_1 - a_2)^2}\right) \\ a_1 + f_{i,j,k} \end{cases}.$$

The three parts correspond to the following intervals, respectively

$$\begin{gathered} f^2_{i,j,k} \geq (a_1 - a_3)^2 + (a_2 - a_3)^2 \\ (a_1 - a_2)^2 \leq f^2_{i,j,k} < (a_1 - a_3)^2 + (a_2 - a_3)^2 \\ f^2_{i,j,k} < (a_1 - a_2)^2. \end{gathered}$$

To solve (B.8), which is not in analytical form, the fast Gauss-Seidel iteration with alternating sweeping orderings is used. For initialisation, the value of grid points on Γ is set to zero, and this value is fixed in later calculations. The rest of the points are set to

large values, and these values will be updated later. The whole 3D grid is traversed in the following orders for the Gauss-Seidel iteration

$$(1)\ i = 1 : M, j = 1 : N, k = 1 : H;\ (2)\ i = M : 1, j = N : 1, k = H : 1;$$
$$(3)\ i = M : 1, j = 1 : N, k = 1 : H;\ (4)\ i = 1 : M, j = N : 1, k = H : 1;$$
$$(5)\ i = M : 1, j = N : 1, k = 1 : H;\ (6)\ i = 1 : M, j = 1 : N, k = H : 1;$$
$$(7)\ i = 1 : M, j = N : 1, k = 1 : H;\ (8)\ i = M : 1, j = 1 : N, k = H : 1.$$

References

[1] N Valous, B Lahrmann, W Zhou, R Veltkamp, and N Grabe. Multistage histopathological image segmentation of iba1-stained murine microglias in a focal ischemia model: Methodological workflow and expert validation. *Journal of Neuroscience Methods*, 213(2):250–262, 2013.

[2] H Zhao, S Osher, and R Fedkiw. Fast surface reconstruction using the level set method. In *Variational and Level set Methods in Computer Vision, 2001. proceedings. IEEE workshop on*, pages 194–201. IEEE, 2001.

[3] G Aubert and P Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, volume 147. Springer Science & Business Media, 2006.

[4] E Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[5] J Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.

[6] D Bertsekas. A simple and fast label correcting algorithm for shortest paths. *Networks*, 23(8):703–709, 1993.

[7] D Bertsekas, F Guerriero, and R Musmanno. Parallel asynchronous label-correcting methods for shortest paths. *Journal of Optimization Theory and Applications*, 88(2):297–320, 1996.

[8] J Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[9] J Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge University Press, 1999.

[10] H Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250):603–627, 2005.

[11] Y Tsai, L Cheng, S Osher, and H Zhao. Fast sweeping algorithms for a class of hamilton–jacobi equations. *SIAM Journal on Numerical Analysis*, 41(2):673–694, 2003.

[12] A Chacon and A Vladimirsky. Fast two-scale methods for eikonal equations. *SIAM Journal on Scientific Computing*, 34(2):A547–A578, 2012.

[13] A Chacon and A Vladimirsky. A parallel two-scale method for eikonal equations. *SIAM Journal on Scientific Computing*, 37(1):A156–A180, 2015.

[14] O Scherzer and J Weickert. Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12(1):43–63, 2000.

[15] L Rudin, S Osher, and E Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[16] P Perona and J Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[17] A Chambolle and T Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[18] T Goldstein, M Li, X Yuan, E Esser, and R Baraniuk. Adaptive primal-dual hybrid gradient methods for saddle-point problems. *Arxiv preprint Arxiv:1305.0546*, 2013.

[19] A Beck and M Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[20] T Goldstein, C Studer, and R Baraniuk. A field guide to forward-backward splitting with a fasta implementation. *Arxiv preprint Arxiv:1411.3406*, 2014.

[21] T Goldstein and S Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[22] K Papafitsoros, C Schoenlieb, and B Sengul. Combined first and second order total variation inpainting using split bregman. *Image Processing on Line*, 3:112–136, 2013.

[23] K Papafitsoros and C Schönlieb. A combined first and second order variational approach for image reconstruction. *Journal of Mathematical Imaging and Vision*, 48(2):308–338, 2014.

[24] W Lu, J Duan, Z Qiu, Z Pan, R Liu, and L Bai. Implementation of high-order variational models made easy for image processing. *Mathematical Methods in the Applied Sciences*, 39(14):4208–4233, 2016.

[25] J Duan, Z Qiu, W Lu, G Wang, Z Pan, and L Bai. An edge-weighted second order variational model for image decomposition. *Digital Signal Processing*, 49:162–181, 2016.

[26] C Wu and X Tai. Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models. *SIAM Journal on Imaging Sciences*, 3(3):300–339, 2010.

[27] X Tai, J Hahn, and G Chung. A fast algorithm for euler's elastica model using augmented lagrangian method. *SIAM Journal on Imaging Sciences*, 4(1):313–344, 2011.

[28] J Hahn, C Wu, and X Tai. Augmented lagrangian method for generalized tv stokes model. *Journal of Scientific Computing*, 50(2):235–264, 2012.

[29] Z Xu, M Figueiredo, and T Goldstein. Adaptive admm with spectral penalty parameter selection. *Arxiv preprint Arxiv:1605.07246*, 2016.

[30] World health organization. *World malaria report 2015*. World health organization, 2016.

[31] A Dondorp, B Angus, K Chotivanich, K Silamut, R Ruangveerayuth, M Hardeman, P Kager, J Vreeken, and N White. Red blood cell deformability as a predictor of anemia in severe falciparum malaria. *The American Journal of Tropical Medicine and Hygiene*, 60(5):733–737, 1999.

[32] L Lawson, V Perry, and S Gordon. Turnover of resident microglia in the normal adult mouse brain. *Neuroscience*, 48(2):405–415, 1992.

[33] M Chirichello and C Chirichello. A standing ovation for looping: The critics respond. *Childhood Education*, 78(1):2–9, 2001.

[34] P Kapeller, S Ropele, Enzinger, T Lahousen, S Strasser-Fuchs, R Schmidt, and F Fazekas. Discrimination of white matter lesions and multiple sclerosis plaques by short echo quantitative 1 h—magnetic resonance spectroscopy. *Journal of Neurology*, 252(10):1229–1234, 2005.

[35] D Zádori, G Nyiri, A Szőnyi, I Szatmári, F Fülöp, J Toldi, T Freund, L Vécsei, and P Klivényi. Neuroprotective effects of a novel kynurenic acid analogue in a transgenic mouse model of huntington's disease. *Journal of Neural Transmission*, 118(6):865–875, 2011.

[36] H Akiyama, S Barger, S Barnum, B Bradt, J Bauer, G Cole, N Cooper, P Eikelenboom, M Emmerling, and B Fiebich. Inflammation and alzheimer's disease. *Neurobiology of Aging*, 21(3):383–421, 2000.

[37] H Jelinek, A Karperien, A Buchan, and T Bossomaier. Differentiating grades of microglia activation with fractal analysis. *Complexity International*, 12:1–12, 2008.

[38] D Donnelly, J Gensel, D Ankeny, N van Rooijen, and P Popovich. An efficient and reproducible method for quantifying macrophages in different experimental models of central nervous system pathology. *Journal of Neuroscience Methods*, 181(1):36–44, 2009.

[39] C Schneider, W Rasband, and K Eliceiri. Nih image to imagej: 25 years of image analysis. *Nature Methods*, 9(7):671–675, 2012.

[40] D Huang, E Swanson, C Lin, J Schuman, W Stinson, W Chang, M Hee, T Flotte, K Gregory, C Puliafito, and J Fujimoto. Optical coherence tomography. *Science*, 254(5035):1178–1181, 1991.

[41] T Chan and J Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.

[42] N Paragios, Y Chen, and O Faugeras. *Handbook of mathematical models in computer vision*. Springer Science & Business Media, 2006.

[43] O Scherzer. *Handbook of mathematical methods in imaging: vol. 1*. Springer Science & Business Media, 2011.

[44] Y You and M Kaveh. Fourth-order partial differential equations for noise removal. *Image Processing, IEEE Transactions on*, 9(10):1723–1730, 2000.

[45] M Lysaker, A Lundervold, and X Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *Image Processing, IEEE Transactions on*, 12(12):1579–1590, 2003.

[46] O Scherzer. Denoising with higher order derivatives of bounded variation and an application to parameter estimation. *Computing*, 60(1):1–27, 1998.

[47] W Hinterberger and O Scherzer. Variational methods on the space of functions of bounded hessian for convexification and denoising. *Computing*, 76(1):109–133, 2006.

[48] R Lai, X Tai, and T Chan. A ridge and corner preserving model for surface restoration. *SIAM Journal on Scientific Computing*, 35(2):A675–A695, 2013.

[49] M Bergounioux and L Piffet. A second-order model for image denoising. *Set-valued and Variational Analysis*, 18(3-4):277–306, 2010.

[50] S Zheng, Z Pan, C Jiang, and G Wang. A new fast algorithm for image denoising. 2013.

[51] G Wang, J Xu, Q Dong, and Z Pan. Active contour model coupling with higher order diffusion for medical image segmentation. *International Journal of Biomedical Imaging*, 2014, 2014.

[52] R Chan, H Liang, S Wei, M Nikolova, and X Tai. High-order total variation regularization approach for axially symmetric object tomography from a single radiograph. *Inverse Problems and Imaging*, 9(1):55–77, 2015.

[53] K Bredies, K Kunisch, and T Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[54] B Goldluecke and D Cremers. Introducing total curvature for image processing. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1267–1274. IEEE, 2011.

[55] W Zhu and T Chan. Image denoising using mean curvature of image surface. *SIAM Journal on Imaging Sciences*, 5(1):1–32, 2012.

[56] W Zhu, X Tai, and T Chan. Augmented lagrangian method for a mean curvature based image denoising model. *Inverse Problems and Imaging*, 7(4):1409–1432, 2013.

[57] X Tai. Fast numerical schemes related to curvature minimization: a brief and elementary review. *Courbure Discrète: Théorie et Applications*, page 17, 2013.

[58] M Nitzberg and D Mumford. The 2.1-d sketch. In *Computer Vision, 1990. Proceedings, third International Conference on*, pages 138–144. IEEE, 1990.

[59] S Masnou and J Morel. Level lines based disocclusion. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 259–263. IEEE, 1998.

[60] T Chan and J Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.

[61] S Esedoglu and J Shen. Digital inpainting based on the mumford–shah–euler image model. *European Journal of Applied Mathematics*, 13(04):353–370, 2002.

[62] W Zhu, T Chan, and S Esedoglu. Segmentation with depth: a level set approach. *SIAM Journal on Scientific Computing*, 28(5):1957–1973, 2006.

[63] S Kang, W Zhu, and J Jianhong. Illusory shapes via corner fusion. *SIAM Journal on Imaging Sciences*, 7(4):1907–1936, 2014.

[64] J Duan, Z Pan, B Zhang, W Liu, and X Tai. Fast algorithm for color texture image inpainting using the non-local ctv model. *Journal of Global Optimization*, 62(4):853—876, 2015.

[65] J Duan, Z Pan, X Yin, W Wei, and G Wang. Some fast projection methods based on chan-vese model for image segmentation. *EURASIP Journal on Image and Video Processing*, 2014(1):1–16, 2014.

[66] J Duan, W Lu, C Tench, I Gottlob, F Proudlock, N Samani, and L Bai. Denoising optical coherence tomography using second order total generalized variation decomposition. *Biomedical Signal Processing and Control*, 24:120–127, 2016.

[67] Y Wang, J Yang, W Yin, and Y Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[68] J Duan, C Tench, I Gottlob, F Proudlock, and L Bai. New variational image decomposition model for simultaneously denoising and segmenting optical coherence tomography images. *Physics in Medicine and Biology*, 60(22):8901, 2015.

[69] J Duan, Y Ding, Z Pan, J Yang, and L Bai. Second order mumford-shah model for image denoising. In *Image processing (ICIP), 2015 IEEE International Conference on*, pages 547–551. IEEE, 2015.

[70] P Getreuer. Rudin-osher-fatemi total variation denoising using split bregman. *Image Processing on Line*, 2:74–95, 2012.

[71] P Getreuer. Rudin-osher-fatemi total variation denoising using split bregman. *Image Processing on Line*, 10, 2012.

[72] P Getreuer. Total variation deconvolution using split bregman. *Image Processing on Line*, 2(1):158–174, 2012.

[73] A Dondorp, F Nosten, P Yi, D Das, A Phyo, J Tarning, K Lwin, F Ariey, W Hanpithakpong, and S Lee. Artemisinin resistance in plasmodium falciparum malaria. *New England Journal of Medicine*, 361(5):455–467, 2009.

[74] H Noedl, D Socheat, and W Satimai. Artemisinin-resistant malaria in asia. *New England Journal of Medicine*, 361(5):540–541, 2009.

[75] W Peters. Drug resistance in plasmodium bergheivincke and lips, 1948. i. chloroquine resistance. *Experimental Parasitology*, 17(1):80–89, 1965.

[76] N Tangpukdee, C Duangdee, P Wilairatana, and S Krudsood. Malaria diagnosis: a brief review. *The Korean Journal of Parasitology*, 47(2):93, 2009.

[77] D Payne. Use and limitations of light microscopy for diagnosing malaria at the primary health care level. *Bulletin of the World Health Organization*, 66(5):621, 1988.

[78] C Murray, R Gasser, A Magill, and R Miller. Update on rapid diagnostic testing for malaria. *Clinical Microbiology Reviews*, 21(1):97–110, 2008.

[79] P Gascoyne, J Satayavivad, and M Ruchirawat. Microfluidic approaches to malaria detection. *Acta Tropica*, 89(3):357–369, 2004.

[80] L Erdman and K Kain. Molecular diagnostic and surveillance tools for global malaria control. *Travel Medicine and Infectious Disease*, 6(1):82–99, 2008.

[81] F Boray Tek, Andrew G Dempster, and Izzet Kale. Parasite detection and identification for automated thin blood film malaria diagnosis. *Computer Vision and Image Understanding*, 114(1):21–32, 2010.

[82] D Memeu, K Kaduki, A Mjomba, N Muriuki, and L Gitonga. Detection of plasmodium parasites from images of thin blood smears. *Open Journal of Clinical Diagnostics*, 3(04):183, 2013.

[83] M Tsai, S Yu, Y Chan, and C Jen. Blood smear image based malaria parasite and infected-erythrocyte detection and segmentation. *Journal of Medical Systems*, 39(10):1, 2015.

[84] N Ross, C Pritchard, D Rubin, and A Duse. Automated image processing method for the diagnosis and classification of malaria on thin blood smears. *Medical and Biological Engineering and Computing*, 44(5):427–436, 2006.

[85] J Somasekar and B Reddy. Segmentation of erythrocytes infected with malaria parasites for the diagnosis using microscopy imaging. *Computers & Electrical Engineering*, 45:336–351, 2015.

[86] S Moon, S Lee, H Kim, L Freitas, M Kang, L Ayong, and M Hansen. An image analysis algorithm for malaria parasite stage classification and viability quantification. *PloS One*, 8(4):e61812, 2013.

[87] S Duffy and V Avery. Development and optimization of a novel 384-well anti-malarial imaging assay validated for high-throughput screening. *The American Journal of Tropical Medicine and Hygiene*, 86(1):84–92, 2012.

[88] W Trager and J Jensen. Human malaria parasites in continuous culture. *Science*, 193(4254):673–675, 1976.

[89] C Lambros and J Vanderberg. Synchronization of plasmodium falciparum erythrocytic stages in culture. *The Journal of Parasitology*, pages 418–420, 1979.

[90] R Liu, L Shi, W Huang, J Xu, S Yu, and D Wang. Generalized total variation-based mri rician denoising model with spatially adaptive regularization parameters. *Magnetic Resonance Imaging*, 32(6):702–720, 2014.

[91] F Knoll, K Bredies, T Pock, and R Stollberger. Second order total generalized variation (tgv) for mri. *Magnetic Resonance in Medicine*, 65(2):480–491, 2011.

[92] Y Ding, M Pardon, A Agostini, H Faas, J Duan, W Ward, F Easton, D Auer, and L Bai. Novel methods for microglia segmentation, feature extraction and classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016.

[93] J Duan, C Tench, I Gottlob, F Proudlock, and L Bai. Automated segmentation of retinal layers from optical coherent tomography images using geodesic distance. *Arxiv preprint Arxiv:1609.02214*, 2016.

[94] D Yang, G Subramanian, J Duan, S Gao, L Bai, R Chandramohanadas, and Y Ai. A portable image-based cytometer for rapid malaria detection and quantification. *PloS One*, 12(6):e0179161, 2017.

[95] A Dowding, A Maggs, and J Scholes. Diversity amongst the microglia in growing and regenerating fish cns: immunohistochemical characterization using fl. 1, an anti-macrophage monoclonal antibody. *Glia*, 4(4):345–364, 1991.

[96] X Luo and S Chen. The changing phenotype of microglia from homeostasis to disease. *Transl Neurodegener*, 1(9), 2012.

[97] B Faith. Microglia: a standing ovation, please! *Brain Sense*, 2011.

[98] L Frick, K Williams, and C Pittenger. Microglial dysregulation in psychiatric disease. *Journal of Immunology Research*, 2013, 2013.

[99] J Vinet, H Weering, A Heinrich, R Kälin, A Wegner, N Brouwer, F Heppner, N Rooijen, H Boddeke, and K Biber. Neuroprotective function for ramified microglia in hippocampal excitotoxicity. *J Neuroinflammation*, 9(1):27, 2012.

[100] L Watkins, E Milligan, and S Maier. Glial activation: a driving force for pathological pain. *Trends in Neurosciences*, 24(8):450–455, 2001.

[101] A Karperien, H Ahammer, and H Jelinek. Quantitating the subtleties of microglial morphology with fractal analysis. *Frontiers in Cellular Neuroscience*, 7, 2013.

[102] J Henkel, D Beers, W Zhao, and S Appel. Microglia in als: the good, the bad, and the resting. *Journal of Neuroimmune Pharmacology*, 4(4):389–398, 2009.

[103] G Kreutzberg. Microglia, the first line of defence in brain pathologies. *Arzneimittelforschung*, 45(3A):357–360, 1995.

[104] W Streit. Microglial response to brain injury: a brief synopsis. *Toxicologic Pathology*, 28(1):28–30, 2000.

[105] H Stanley, L Amaral, A Goldberger, S Havlin, P Ivanov, and C Peng. Statistical physics and physiology: monofractal and multifractal approaches. *Physica A: Statistical Mechanics and its Applications*, 270(1):309–324, 1999.

[106] L Cysique, K Moffat, D Moore, T Lane, N Davies, A Carr, B Brew, and C Rae. Hiv, vascular and aging injuries in the brain of clinically stable hiv-infected adults: a 1h mrs study. *PloS One*, 8(4):e61738, 2013.

[107] D Mumford and J Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[108] J Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Computer vision and Pattern Recognition, 1996. proceedings CVPR'96, 1996 IEEE computer society conference on*, pages 136–142. IEEE, 1996.

[109] J Duan, J Yang, and L Bai. Second order mumford-shah model for image denoising. In *International conference on image processing*. IEEE, 2015.

[110] C Kozlowski and R Weimer. An automated method to quantify microglia morphology and application to monitor activation state longitudinally in vivo. *PloS One*, 7(2):e31814, 2012.

[111] I Bertha Hovens, Csaba N, and Regien G S. A novel method for evaluating microglial activation using ionized calcium-binding adaptor protein-1 staining: cell body to cell size ratio. *Neuroimmunology and Neuroinflammation*, 1(2):82, 2014.

[112] Z Sołtys, M Ziaja, R Pawliński, Z Setkowicz, and K Janeczko. Morphology of reactive microglia in the injured cerebral cortex. fractal analysis and complementary quantitative methods. *Journal of Neuroscience Research*, 63(1):90–97, 2001.

[113] T Wyss-Coray and J Rogers. Inflammation in alzheimer disease—a brief review of the basic science and clinical literature. *Cold Spring Harbor Perspectives in Medicine*, 2(1):a006346, 2012.

[114] D Cremers, F Tischhäuser, J Weickert, and C Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.

[115] G Aubert, M Barlaud, O Faugeras, and S Jehan. Image segmentation using active contours: Calculus of variations or shape gradients? *SIAM Journal on Applied Mathematics*, 63(6):2128–2154, 2003.

[116] A Herbulot, S Jehan, S Duffner, M Barlaud, and Gi Aubert. Segmentation of vectorial image features using shape gradients and information measures. *Journal of Mathematical Imaging and Vision*, 25(3):365–386, 2006.

[117] E Debreuve, M Gastaud, M Barlaud, and G Aubert. Using the shape gradient for active contour segmentation: from the continuous to the discrete formulation. *Journal of Mathematical Imaging and Vision*, 28(1):47–66, 2007.

[118] S Osher and J Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[119] H Zhao, T Chan, B Merriman, and S Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996.

[120] T Chan and L Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.

[121] L Vese and T Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[122] G Chung and L Vese. Energy minimization based segmentation and denoising using a multilayer level set approach. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 439–455. Springer, 2005.

[123] C Liu, Z Pan, and J Duan. New algorithm for level set evolution without re-initialization and its application to variational image segmentation. *Journal of Software*, 8(9):2305–2312, 2013.

[124] J Lie, M Lysaker, and X Tai. A binary level set model and some applications to mumford-shah image segmentation. *Image Processing, IEEE Transactions on*, 15(5):1171–1181, 2006.

[125] J Lie, M Lysaker, and X Tai. A variant of the level set method and applications to image segmentation. *Mathematics of Computation*, 75(255):1155–1174, 2006.

[126] X Bresson, S Esedoglu, P Vandergheynst, and J Thiran. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, 2007.

[127] F Li, M Ng, T Zeng, and C Shen. A multiphase image segmentation method based on fuzzy region competition. *SIAM Journal on Imaging Sciences*, 3(3):277–299, 2010.

[128] F Li, S Osher, J Qin, and M Yan. A multiphase image segmentation based on fuzzy membership functions and l1-norm fidelity. *Arxiv preprint Arxiv:1504.02206*, 2015.

[129] L Ambrosio and V Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[130] J Duan, W Lu, Z Pan, and L Bai. New second order mumford-shah model based on γ-convergence approximation for image processing. *Infrared Physics and Technology*, 76:641–647, 2016.

[131] J Duan, C Tench, I Gottlob, F Proudlock, and L Bai. Optical coherence tomography image segmentation. In *Image processing (ICIP), 2015 IEEE International Conference on*, pages 4278–4282. IEEE, 2015.

[132] C Bonardi, F Pulford, D Jennings, and M Pardon. A detailed analysis of the early context extinction deficits seen in appswe/ps1de9 female mice and their relevance to preclinical alzheimer's disease. *Behavioural Brain Research*, 222(1):89–97, 2011.

[133] H Ishikawa, D Stein, G Wollstein, S Beaton, J Fujimoto, and J Schuman. Macular segmentation with optical coherence tomography. *Investigative Ophthalmology & Visual Science*, 46(6):2012–2017, 2005.

[134] M Shahidi, Z Wang, and R Zelkha. Quantitative thickness measurement of retinal layers imaged by optical coherence tomography. *American Journal of Ophthalmology*, 139(6):1056–1061, 2005.

[135] D Fernández, H Salinas, and C Puliafito. Automated detection of retinal layer structures on optical coherence tomography images. *Optics Express*, 13(25):10200–10216, 2005.

[136] M Mayer, R Tornow, R Bock, J Hornegger, and F Kruse. Automatic nerve fiber layer segmentation and geometry correction on spectral domain oct images using fuzzy c-means clustering. *Investigative Ophthalmology and Visual Science*, 49(13):1880–1880, 2008.

[137] A Mishra, A Wong, K Bizheva, and D Clausi. Intra-retinal layer segmentation in optical coherence tomography images. *Optics Express*, 17(26):23719–23728, 2009.

[138] A Yazdanpanah, G Hamarneh, B Smith, and M Sarunic. Intra-retinal layer segmentation in optical coherence tomography using an active contour approach. In *Medical image computing and computer-assisted intervention–MICCAI 2009*, pages 649–656. Springer, 2009.

[139] I Ghorbel, F Rossant, I Bloch, S Tick, and M Paques. Automated segmentation of macular layers in oct images and quantitative evaluation of performances. *Pattern Recognition*, 44(8):1590–1603, 2011.

[140] F Rossant, I Bloch, I Ghorbel, and M Paques. Parallel double snakes. application to the segmentation of retinal layers in 2d-oct for pathological subjects. *Pattern Recognition*, 48(12):3857–3870, 2015.

[141] S Chiu, X Li, P Nicholas, C Toth, J Izatt, and S Farsiu. Automatic segmentation of seven retinal layers in sdoct images congruent with expert manual segmentation. *Optics Express*, 18(18):19413–19428, 2010.

[142] Q Yang, C Reisman, Z Wang, Y Fukuma, M Hangai, N Yoshimura, A Tomidokoro, M Araie, A Raza, and D Hood. Automated layer segmentation of macular oct images using dual-scale gradient information. *Optics Express*, 18(20):21293–21307, 2010.

[143] X Liu, J Wang, Z Yang, and W Hu. Automated segmentation of nine retinal layers with layer thickness information on sd-oct images. In *Eighth International Conference on Digital Image Processing (ICDIP 2016)*, pages 100331G–100331G. International society for optics and photonics, 2016.

[144] V Kajić, B Považay, B Hermann, B Hofer, D Marshall, P Rosin, and W Drexler. Robust segmentation of intraretinal layers in the normal human fovea using a novel statistical model based on texture and shape analysis. *Optics Express*, 18(14):14730–14744, 2010.

[145] V Kajić, M Esmaeelpour, B Považay, D Marshall, P Rosin, and W Drexler. Automated choroidal segmentation of 1060 nm oct in healthy and pathologic eyes using a statistical model. *Biomedical Optics Express*, 3(1):86–103, 2012.

[146] M Pilch, Y Wenner, E Strohmayr, M Preising, C Friedburg, E Bexten, B Lorenz, and K Stieger. Automated segmentation of retinal blood vessels in spectral domain optical coherence tomography scans. *Biomedical Optics Express*, 3(7):1478–1491, 2012.

[147] M Mujat, R Chan, B Cense, B Park, C Joo, T Akkin, T Chen, and J Boer. Retinal nerve fiber layer thickness map determined from optical coherence tomography images. *Optics Express*, 13(23):9480–9491, 2005.

[148] T Cootes, C Taylor, D Cooper, and J Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[149] T Cootes, G Edwards, and C Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):681–685, 2001.

[150] M Garvin, M Abràmoff, X Wu, S Russell, T Burns, and M Sonka. Automated 3-d intraretinal layer segmentation of macular spectral-domain optical coherence tomography images. *Medical Imaging, IEEE Transactions on*, 28(9):1436–1447, 2009.

[151] G Quellec, K Lee, M Dolejsi, M Garvin, M Abràmoff, and M Sonka. Three-dimensional analysis of retinal layer texture: identification of fluid-filled regions in sd oct of the macula. *Medical Imaging, IEEE Transactions on*, 29(6):1321–1330, 2010.

[152] B Antony, M Abràmoff, M Sonka, Y Kwon, and M Garvin. Incorporation of texture-based features in optimal graph-theoretic approach with application to the 3d segmentation of intraretinal surfaces in sd-oct volumes. In *SPIE Medical Imaging*, pages 83141G–83141G. International Society for Optics and Photonics, 2012.

[153] P Dufour, L Ceklic, H Abdillahi, S Schroder, S De Dzanet, U Wolf-Schnurrbusch, and J Kowal. Graph-based multi-surface segmentation of oct data using trained hard and soft constraints. *Medical Imaging, IEEE Transactions on*, 32(3):531–543, 2013.

[154] R Kafieh, H Rabbani, M Abramoff, and M Sonka. Intra-retinal layer segmentation of 3d optical coherence tomography using coarse grained diffusion map. *Medical Image Analysis*, 17(8):907–928, 2013.

[155] J Tian, B Varga, G Somfai, W Lee, W Smiddy, and D DeBuc. Real-time automatic segmentation of optical coherence tomography volume data of the macular region. *PloS One*, 10(8):e0133908, 2015.

[156] J Tian, B Varga, E Tatrai, P Fanni, G Somfai, W Smiddy, and D Debuc. Performance evaluation of automated segmentation software on optical coherence tomography volume data. *Journal of Biophotonics*, 9(5):478–489, 2016.

[157] K Vermeer, J Schoot, H Lemij, and J Boer. Automated segmentation by pixel classification of retinal layers in ophthalmic oct images. *Biomedical Optics Express*, 2(6):1743–1756, 2011.

[158] A Fuller, R Zawadzki, S Choi, D Wiley, J Werner, and B Hamann. Segmentation of three-dimensional retinal image data. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1719–1726, 2007.

[159] M Szkulmowski, M Wojtkowski, B Sikorski, T Bajraszewski, V Srinivasan, A Szkulmowska, J Kałużny, J Fujimoto, and A Kowalczyk. Analysis of posterior retinal layers in spectral optical coherence tomography images of the normal retina and retinal pathologies. *Journal of Biomedical Optics*, 12(4):041207–041207, 2007.

[160] A Lang, A Carass, M Hauser, E Sotirchos, P Calabresi, H Ying, and J Prince. Retinal layer segmentation of macular oct images using boundary classification. *Biomedical Optics Express*, 4(7):1133–1152, 2013.

[161] J Duan, B Haines, W Ward, and L Bai. Surface reconstruction from point clouds using a novel variational model. In *Research and Development in Intelligent Systems XXXII*, pages 135–146. Springer, 2015.

[162] D DeBuc. *A review of algorithms for segmentation of retinal image data using optical coherence tomography.*

[163] Q Song, X Wu, Y Liu, M Sonka, and M Garvin. Simultaneous searching of globally optimal interacting surfaces with shape priors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE conference on*, pages 2879–2886. IEEE, 2010.

[164] J Duan, Z Qiu, W Lu, G Wang, Z Pan, and L Bai. An edge-weighted second order variational model for image decomposition. *Digital Signal Processing*, 49:162–181, 2016.

[165] H Edelsbrunner and E Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72, 1994.

[166] N Amenta, M Bern, and M Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 415–421. ACM, 1998.

[167] J Ye, X Bresson, T Goldstein, and S Osher. A fast variational method for surface reconstruction from sets of scattered points. *CAM report*, 10(01), 2010.

[168] J Liang, F Park, and H Zhao. Robust and efficient implicit surface reconstruction for point clouds based on convexified image segmentation. *Journal of Scientific Computing*, 54(2-3):577–602, 2013.

[169] S Osher and J Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[170] H Zhao, S Osher, B Merriman, and M Kang. Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding*, 80(3):295–314, 2000.

[171] C Li, C Xu, C Gui, and M Fox. Level set evolution without re-initialization: a new variational formulation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 430–436. IEEE, 2005.

[172] M Kazhdan, M Bolitho, and H Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics Symposium on Geometry Processing*, 2006.

[173] M Kass, A Witkin, and D Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[174] V Caselles, R Kimmel, and G Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[175] H Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250):603–627, 2005.

[176] T Chan, S Esedoglu, and M Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.

[177] T Chan and L Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.

[178] A Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.

[179] X Bresson and T Chan. Active contours based on chambolle's mean curvature motion. In *Image processing, 2007. ICIP 2007. IEEE International Conference on*, volume 1, pages I–33. IEEE, 2007.

[180] T Goldstein and X Bresson. Geometric applications of the split bregman method: segmentation and surface reconstruction. *Journal of Scientific Computing*, 45(1-3):272–293, 2010.

[181] D Adalsteinsson and J Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

[182] S Osher and R Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

[183] M Sussman, P Smereka, and S Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[184] R Courant, K Friedrichs, and H Lewy. On the partial difference equations of mathematical physics. Technical report, California Univ Los Angeles, 1959.

[185] S Alliney. A property of the minimum vectors of a regularizing functional defined by means of the absolute norm. *IEEE Transactions on Signal Processing*, 45(4):913–917, 1997.

[186] M Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers. *SIAM Journal on Numerical Analysis*, 40(3):965–994, 2002.

[187] M Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1):99–120, 2004.

[188] T Chan and S Esedoglu. Aspects of total variation regularized l1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

[189] T Pock, D Cremers, H Bischof, and A Chambolle. An algorithm for minimizing the mumford-shah functional. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1133–1140. IEEE, 2009.