Variational Models and Fast Numerical Algorithms for Selective Image Segmentation

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy

by

Abdul K. Jumaat

under the supervision of:

Professor Ke Chen

May 2019

Contents

Acknowledgements

Foremost, praise is to the Almighty, on whom ultimately I depend for sustenance and guidance. This thesis is dedicated to my late father Jumaat and my late father in law Azahar.

I would like to express my sincere gratitude to my advisor Prof. Ke Chen for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. Not to forget my second supervisor Dr Kieran Sharkey for his support and encouragement. I also would like to thank Prof. Natasha Movchan as well as Dr Gayane Piliposyan for their advice during my doctoral studies.

My sincere and deepest appreciation to all the CMIT's members including Dr. Bryan Williams, Dr. Jianping Zhang, Dr. Mazlinda Ibrahim, Dr. Jack Spencer, Dr. Anis Theljani, Daoping Zhang, Anthony Thompson, Michael Roberts and Liam Burrows for all the valuable time, advice, criticism and discussion.

I am grateful to my family especially my mother, Mrs Kamariah for her prayers, courage and advice. A very special gratitude goes out to my lovely wife Nursaliza who patiently and lovingly supported me throughout my Ph.D journey and for taking care of our children 'Afif and Aisyah in my absence.

I am very thankful to the government of Malaysia and MARA University of Technology for financial, technical, and academic support. I know that this research would not have been possible without their support. This research paves the way for further exploration of other problems in mathematical imaging field that benefits the sponsors, research community and related industries.

And finally, last but by no means least, also to everyone in the MSC Liverpool, we had a great time in United Kingdom with exciting activities. Indeed, this entire journey helped me to obtain better understanding of many valuable things in this life.

Thanks for all your encouragement!

Abstract

Image segmentation is a fundamental task in image analysis that aims at partitioning an image into sub-regions or at representing the image into something that is more meaningful and easier to analyse. Variational image segmentation models have become very popular in recent years, especially global image segmentation models which aim to segment all objects in an image. Given a set of user-defined prior points, selective variational models aim to segment selectively one object only. Time marching methods with semi-implicit schemes (gradient descents) or additive operator splitting (AOS) are used frequently to solve the resulting partial differential equation (PDE) of Euler Lagrange equations derived from these models. For images of moderate size, such methods are effective. However, to segment images of large size, urgent need exists to develop fast iterative solvers.

We first propose an optimisation based multilevel algorithm for efficiently solving a class of nonconvex selective segmentation models. In literature, the models are originally solved based on optimise-discretise scheme where the PDE derived from the models are numerically solved using AOS method. The multilevel method we use is based on discretise-optimise scheme where minimisation of a variational problem is solved directly without using a PDE. Numerical results on synthetic and real medical data show that good segmentation quality is obtained and, as expected, excellent efficiency is observed in reducing computational time compare to AOS method.

Secondly, we propose a new convex selective segmentation model, allowing a global minimiser to be found independently of initialisation. The existing convex segmentation model however is expensive to solve and suffers from parameter sensitivity due to existence of a highly nonlinear term in the formulation. Our new formulation using primal-dual framework is able to reduce the complexity of the existing model. To speed up the segmentation process, we also develop a multilevel algorithm for the new convex segmentation model. Experiments using synthetic and real medical data shows that the new model is less sensitive to parameters compared to the existing convex model and an optimal computational time is achieved.

Many application fields such as medical imaging, geological surveying and computational fluid dynamics can greatly benefit from 3-D selective segmentation as a 3-D representation carries more information compared to 2-D representation. This information is highly useful for example in medical surgery planning. However, there exist little literature addressing selective segmentation in 3-D and their formulations are nonconvex. Hence, we present an extension of the multilevel algorithm and convex selective variational image segmentation model above into 3-D framework. Numerical tests show that the proposed model is effective and the algorithm is efficient in locally segmenting 3-D complex image structures.

Due to natural complexity of real images, the targeted objects might be occluded by other ones or some parts of them may not be distinguished from the background. For example, in medical image analysis, targeted tumour might be blended by other ones or some part of them may be occluded by other organs or tumour or some object

boundaries even missing due to imaging conditions. The grey intensity selective based segmentation models might not be well suited to do the segmentation task as the models heavily rely on grey intensity values of the given image. In the final study, we develop two new selective segmentation models which impose curvature constraints on the formulations to restore those boundaries that are missing or not well defined by the grey intensity images. On top of that, we develop multilevel algorithms to solve these higher order minimisation problems. Numerical tests demonstrated that the models give satisfactory results in optimum computational time when compared to other existing models.

Publications and Presentations

Publications:

- A. K. Jumaat and K. Chen. An optimization-based multilevel algorithm for variational image segmentation models. Electronic Transactions on Numerical Analysis, 46:474-504, 2017.
- A. K. Jumaat and K. Chen. Fast Algorithm for Selective Image Segmentation Model. International Journal of Engineering & Technology, 4.33:41-44, 2018.
- A. K. Jumaat and K. Chen. A reformulated convex and selective variational image segmentation model and its fast multilevel algorithm. Numerical Mathematics: Theory, Methods and Applications, 12:403-437, 2019.
- A. K. Jumaat and K. Chen. A Three-dimensional Convex and Selective Variational Image Segmentation Model and Its Fast Algorithms. In Preparation. 2019.

Presentations:

- A. K. Jumaat and K. Chen. An optimization-based multilevel algorithm for convex variational image segmentation models, Summer School, 23-27 May 2016, University of Greenwich, London.
- A. K. Jumaat and K. Chen. An Optimization Based Multilevel Algorithm for Selective Variational Image Segmentation Models, 27th Biennial Numerical Analysis Conference, University of Strathclyde, 27-30 June 2017, Glasgow.
- A. K. Jumaat and K. Chen. Fast multilevel algorithms for nonlinear optimization in image processing, SIAM Conference on Imaging Science, 5-8 June 2018, Bologna, Italy.

List of Figures

List of Tables

Chapter 1

Introduction

This thesis will focus on a specific branch of computer vision called image segmentation. The objective of image segmentation is to obtain meaningful partitions of an input image into a finite number of disjoint homogeneous regions. This task is obviously very difficult to achieve and there exist many methods to realize it. The definition of meaningful depends on the problem setting, and the possible application. For example, it could mean identifying the boundary of an organ or tumour in medical imaging field. From a security perspective, it might mean selecting certain objects such as vehicles or people from the image.

Different models and techniques have been developed so far, including histogram analysis and thresholding [88, 121], region growing [1], edge detection and active contours [7, 38]. Of all these techniques, variational techniques [38, 102] have proven to be very efficient for extracting homogeneous areas compared with other models such as statistical methods [45, 44, 58] or wavelet techniques [90].

Segmentation models can be classified into two categories, namely, edge based and region based models; other models may mix these categories. Edge based models refer to the models that are able to drive the contours towards image edges by influence of an edge detector function used. The snake algorithm proposed by Kass et al. [77] was the first edge based variational models for image segmentation. Further improvement on the algorithm with Geodesic Active Contours and the level-set formulation led to effective models [25, 123].

Region based segmentation techniques try to separate all pixels of an object from its background pixels based on the intensity and hence find image edges between regions satisfying different homogeneity criteria. Examples of region-based techniques are Region growing [69, 16], Watershed algorithm [69], Thresholding [69, 143], and Fuzzy clustering [126] and variational models such as Mumford-Shah [102] model and the Chan-Vese (CV) [38] model.

The most celebrated and efficient variational model for the images with and without noise is the Mumford-Shah [102] model, that reconstructs the segmented image as a piecewise smooth intensity function. Since the model cannot be implemented directly and easily, it is often approximated. The Chan-Vese (CV) [38] model is simplified and reduced from [102], without approximation. The simplification is to replace the

piecewise smooth function by a piecewise constant function and, in the case of two phases, the piecewise constant function divides an image into the foreground and the background.

Segmentation models described above are used for global segmentation due to the fact that all features or objects in an image are to be segmented. This thesis is concerned with another type of image segmentation models, namely selective segmentation models. They are defined as the process of extracting one object of interest in an image based on some known geometric constraints [65, 111, 129].

Two effective models are Badshah-Chen [12] and Rada-Chen [111] which use a mixture of edge-based and region-based ideas in addition to imposing constraints. Recently, a convex selective variational image segmentation model called as Convex Distance Selective Segmentation was successfully proposed by Spencer and Chen [129]. The convex model allows a global minimiser to be found independently of initialisation [129, 34]. The additive operator splitting (AOS) method was proposed to solve the models which is suitable for image of moderate size. However, to process images of large size, urgent need exists in developing fast methods.

The multigrid and multilevel method have been developed using the idea of hierarchy of discretisations to solve problem with huge data. A multilevel method is based on discretise-optimise scheme where minimisation of a variational problem is solved directly without using a PDE. In contrast, a multigrid method is based on optimise-discretise scheme where it solves a resultant PDE generally from Euler Lagrange equation derived from the objective functional. The two methods are inter-connected since both can have geometric interpretations and use similar inter-level information transfers.

The multigrid methods have been used to solve a few variational image segmentation models in the level set formulation [78, 108, 109, 10, 11]. While the practical performance of these methods is good, however, they are sensitive to parameters and hence not effective, mainly due to non-smooth coefficients which lead to smoothers not having an acceptable smoothing rate and behave like the cascadic multigrids [100] where only one multigrid cycle is needed.

Here we pursue multilevel optimisation methods, based on a discretise-optimise scheme where the minimisation is solved directly (without using PDEs). The idea has been applied to other image problems in denoising and debluring [33, 29, 30], but not yet to selective segmentation problems.

1.1 Thesis Outline

The remaining chapters of this thesis are organised as follows.

Chapter 2 - Mathematical Preliminaries

In this chapter, we introduce some mathematical tools which will be used throughout this thesis. A brief review is given on definitions, theorems and examples of some important and relevant mathematical topics including normed vector spaces, convex

sets and functions, calculus of variations, and functions of bounded variations. In relation to variational methods, we also discuss inverse problems and regularisation, discretising partial differential equations, image representation, level set method and iterative solutions to equations.

Chapter 3 - Review of Variational Models in Image Processing

In this chapter, the most important segmentation models in variational setting are reviewed. These are useful and closely related to our work. The models include Mumford-Shah model [102], Snake model [77], Geodesic Active Contour [25] , Chan and Vese model [38], global minimisation model [34], Geodesic Aided CV Model [41], and lastly the selective segmentation model namely Geometrical Constraint Segmentation Model [67].

Chapter 4 - An Optimisation based Multilevel Algorithm for Variational Image Segmentation Models

In this chapter we develop an optimisation based multilevel algorithm for efficiently solving two nonconvex selective segmentation models. Moreover, we propose the modified localised version of these models to exploit the local nature of segmentation contours in order to speed up the computation time, solved using multilevel algorithm. Experiment is done using synthetic and real medical data.

Chapter 5 - A Reformulated Convex and Selective Variational Image Segmentation Model and its Fast Multilevel Algorithm

The existing convex segmentation model is expensive to solve and suffers from parameter sensitivity due to existence of highly nonlinear term in the formulation, hence reformulated version of the model using primal-dual framework is done in this chapter. In addition, we also develop a multilevel algorithm to solve both the original and the new model. To get a stronger decaying property, a new variant of the multilevel algorithm to solve the new model is introduced where we can prove the convergence of the multilevel algorithm.

Chapter 6 - A Three-dimensional Convex and Selective Variational Image Segmentation Model and Its Fast Algorithms

Developing a reliable 3-D segmentation model is important to many fields of study as it can produce results with much information for further analysis compare to 2-D model. We introduce a 3-D convex segmentation model in this chapter. Localised version of the model is introduced to deal with large 3-D data, solved using a new develop 3-D multilevel framework to ensure the computational speed is optimum. A new variant for 3-D multilevel algorithm to solve the 3-D model is proposed and its convergence is proven.

Chapter 7 - Euler's Elastica based Selective Segmentation Model and Its Fast Algorithm

Some objects in images might be occluded by other ones or some parts of them may not be distinguishable from the background. Consequently, the segmentation task is extremely hard and may not be possible for grey intensity based segmentation models. To do this segmentation task, we propose two new Euler's Elastica based selective segmentation models which impose curvature constraints on the formulation. These restore those boundaries that are missing or not well defined. Multilevel algorithm is proposed to solve these higher order minimisation problems efficiently.

Chapter 8 - Conclusion and Future Research

In the final chapter, we discuss our conclusions and outline possible future research directions arising from the work presented in this thesis.

Chapter 2

Mathematical Preliminaries

In this chapter we provide a summary of relevant mathematical preliminaries that will be used in later chapters. We first introduce some concepts of normed linear space and some background for functions of bounded variation. Then, the idea about inverse problem and the theory of calculus of variations are introduced before moving on to discretisation of partial differential equations (PDE) related to inverse problem and concepts on interface representation. Next, we provide an overview of conventional methods for iteratively solving equations, both in the linear and nonlinear case. We end this chapter with an introduction to multigrid and optimisation based multilevel algoritms.

2.1 Normed Vector Spaces

We first introduce the vector space (also called a linear space), a basic mathematical structure formed by a collection of elements

$$\mathbf{u} = (u_1, ..., u_n), \quad \mathbf{v} = (v_1, ..., v_n)$$

called vectors. Literature can commonly be found in either linear algebra or advanced calculus literature such as [5, 79, 118].

Definition 2.1.1. *(**Vector Space**). Let* $\mathbf{V}$ *be a vector set on which two operations, addition and scalar multiplications, have been defined. For* $\mathbf{u}, \mathbf{v} \in \mathbf{V}$*, the sum of* $\mathbf{u}$ *and* $\mathbf{v}$ *is denoted by* $\mathbf{u} + \mathbf{v}$*, and if* c *is a scalar, the scalar multiple of* $\mathbf{u}$ *by* c *is denoted by* $c\mathbf{u}$*. If the following axioms hold for all* $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{V}$ *and for all scalars* c, d*, then* $\mathbf{V}$ *is called a vector space and its elements are called vectors.*

1. *Closure under addition: If* $\mathbf{u}, \mathbf{v} \in \mathbf{V}$*, then* $\mathbf{u} + \mathbf{v} \in \mathbf{V}$*.*
2. *Commutativity under addition: If* $\mathbf{u}, \mathbf{v} \in \mathbf{V}$*, then* $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$*.*
3. *Associativity under addition: If* $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{V}$*, then* $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$*.*
4. *Existence of an identity element of addition: There exists an element* $\mathbf{0} \in \mathbf{V}$*, called a zero vector, such that* $\mathbf{u} + \mathbf{0} = \mathbf{u}$ *for all* $\mathbf{u} \in \mathbf{V}$*.*

5. *Existence of additive inverse: For each* $\mathbf{u} \in \mathbf{V}$*, there is an element* $-\mathbf{u} \in \mathbf{V}$ *such that* $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$.

6. *Closure under scalar multiplication: If* c *is a scalar, and* $\mathbf{u} \in \mathbf{V}$ *, then* $c\mathbf{u} \in \mathbf{V}$.

7. *Distributivity: If* $\mathbf{u}, \mathbf{v} \in \mathbf{V}$ *and* c *is a scalar then* $c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$.

8. *Distributivity under scalar multiplication: If* $\mathbf{u} \in \mathbf{V}$ *and* c, d *are scalar then* $(c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u}$.

9. *Associativity under scalar multiplication: If* $\mathbf{u} \in \mathbf{V}$ *and* c, d *are scalar then* $c(d\mathbf{u}) = (cd)\mathbf{u}$.

10. *Existence of an identity element of scalar multiplication: There exists an element scalar* 1 *called the scalar multiplicative identity, such that* $1\mathbf{u} = \mathbf{u}$ *for all* $\mathbf{u} \in \mathbf{V}$.

Example 2.1.2. *Examples of vector spaces include:*

- *The space* $C^k(\Omega)$ *of all functions on the domain* $\Omega \subset \mathbb{R}^d$ *whose partial derivatives of order up to* k *are continuous.*
- *The space* $\mathbb{R}^d$ *for all* $d \in \mathbb{N}$.

Definition 2.1.3. ***(Norm and Seminorm).*** *A norm* $\|\cdot\|$ *on a vector space* $\mathbf{V}$ *is a real-valued function* $\|\cdot\| : \mathbf{V} \to \mathbb{R}$ *that satisfies the following properties:*

1. *Faithfulness:* $\|\mathbf{u}\| = 0$ *if and only if* $\mathbf{u} = 0$ *and* $\|\mathbf{u}\| > 0$ *if* $\mathbf{u} \neq 0$.
2. *Absolute homogeneity:* $\|\lambda\mathbf{u}\| = |\lambda| \|\mathbf{u}\|$ *for all scalars* λ.
3. *Triangle inequality:* $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$

for all $\mathbf{u}, \mathbf{v} \in \mathbf{V}$.

A seminorm is defined similarly as above but the first property is replaced by $\|\mathbf{u}\| \geq 0$.

Example 2.1.4. *(p-norm). Consider* $\mathbf{x} \in \mathbb{R}^d$ *, then for any real number* $p \geq 1$ *the p-norm of* $\mathbf{x}$ *is defined by*

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}},$$

where for $p = 2$ *we have the Euclidean norm. The infinity norm is defined as*

$$\|\mathbf{x}\|_\infty = \max\left(|x_1|, |x_2|, ..., |x_n|\right).$$

Example 2.1.5. *(L^p-norm). Consider a continuous function* f *defined on a domain* Ω *such that*

$$\int_\Omega |f(\mathbf{x})|^p d\mathbf{x} < \infty,$$

with $1 \leq p \leq \infty$. Then the L^p-norm of f on Ω is defined as

$$\|f(\mathbf{x})\|_{L^p} = \left(\int_\Omega |f(\mathbf{x})|^p d\mathbf{x} \right)^{\frac{1}{p}}.$$

The special case of $p = \infty$ is defined as

$$\|f(\mathbf{x})\|_\infty = \sup_{\mathbf{x}} |f(\mathbf{x})| .$$

Definition 2.1.6. ***(Inner Product)**. Let $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{V}$ with a scalar λ. An inner product on the vector space $\mathbf{V}$ is a function $\langle \cdot, \cdot \rangle_{\mathbf{V}}$ defined on $\mathbf{V} \times \mathbf{V}$ which satisfies*

1. *Positive definiteness: $\langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{V}} > \mathbf{0}$ when $\mathbf{u} \neq \mathbf{0}$.*
2. *Linearity under scalar multiplication: $\langle \lambda \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}} = \lambda \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}}$.*
3. *Linearity under vector addition: $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle_{\mathbf{V}} = \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}} + \langle \mathbf{v}, \mathbf{w} \rangle_{\mathbf{V}}$.*
4. *Symmetry: $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}} = \langle \mathbf{v}, \mathbf{u} \rangle_{\mathbf{V}}$.*

Example 2.1.7. *Examples of inner products are*

- *The standard inner product*

$$\langle \mathbf{x}, \mathbf{y} \rangle_d = \sum_{i=1}^{d} x_i y_i$$

  for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

- *For real-valued continuous functions $f(\mathbf{x})$ and $g(\mathbf{x})$ defined on the interval $[a, b]$*

$$\langle f, g \rangle = \int_a^b f(\mathbf{x}) g(\mathbf{x}) \, d\mathbf{x}$$

When given a complex vector space, the last property above is replaced by conjugate symmetry: $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}} = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}_{\mathbf{V}}$.

Definition 2.1.8. ***(Normed Vector Space)**. If a vector space $\mathbf{V}$ is equipped with a norm $\|\cdot\|$ defined on it, then $\mathbf{V}$ is called a normed vector space (also called a normed linear space).*

Remark 2.1.9. *A relevant example is Euclidean n-space (or Cartesian space), where the space of all n-tuples of real numbers $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is equipped with the Euclidean metric (distance between two elements $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$)*

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} (x_i - y_i) \right)^{\frac{1}{2}}.$$

Any inner product defined on a vector space $\mathbf{V}$ induces a norm defined by $\|\mathbf{u}\|_{\mathbf{V}} = \langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{V}}^{1/2}$, is a special type of normed vector space. When a vector space is equipped with a seminorm, then it is called a seminormed vector space.

Definition 2.1.10. *(**Cauchy Sequence and Completeness**). A sequence* $\{x_i\}_{i\in\mathbb{N}}$ *in a normed vector space* $\mathbf{V}$ *is called a Cauchy sequence if for any real number* $\varepsilon > 0$*, there exists* $M \in \mathbb{Z}^+$ *such that for every natural number* $m, n > M$ *, we have* $\|x_m - x_n\| < \varepsilon$ *. A normed vector space* $\mathbf{V}$ *is said to be complete if every Cauchy sequence* $\{x_i\}_{i\in\mathbb{N}} \in \mathbf{V}$ *converges to an element* $x \in \mathbf{V}$ *which implies that* $\lim_{i\to\infty} x_i = x$*.*

Note that Cauchy sequence is a sequence where all terms, except a counted number, become arbitrarily close to one another.

Definition 2.1.11. *(**Banach Space**). A complete normed vector space is called Banach space.*

Example 2.1.12. *The space of all continuous functions* f *in an interval* $[a, b]$*, denoted* $C([a, b], \mathbb{R})$*, is a Banach space if we define the supremum norm of such function as*

$$\|f\| = \sup\{|f(x)| : x \in [a, b]\}.$$

It is a well-defined norm since all continuous functions on a compact interval are bounded.

Definition 2.1.13. *(**Hilbert Space**). A Hilbert space is a vector space* $\mathbf{V}$ *with an inner product defined on it such that every Cauchy sequence converges to an element of the vector space* $\mathbf{V}$*.*

Example 2.1.14. *Two relevant examples of Hilbert space are the space* $\mathbb{R}^d$ *together with the Euclidean inner product and the space* $L^2(\Omega)$ *together with the inner product defined by*

$$\langle f, g\rangle_{L^2(\Omega)} = \int_\Omega f(\mathbf{x})g(\mathbf{x})\, d\mathbf{x}.$$

Definition 2.1.15. *(**Lipschitz Continuous Function**). A function* J *is called Lipschitz continuous with Lipschitz constant* L_f *on* $\mathbb{R}$ *if there is a nonnegative constant* L_f *such that*

$$\|J(\mathbf{y}) - J(\mathbf{x})\| \leq L_f \|\mathbf{y} - \mathbf{x}\| \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}$$

for any given operator norm.

Definition 2.1.16. *(**Convex Set**). A set* S *in a vector space* $\mathbf{V}$ *is said to be convex if, for all* $\mathbf{u}, \mathbf{v} \in S$ *and all* $\lambda \in [0, 1]$ *, the point*

$$(1 - \lambda)\,\mathbf{u} + \lambda\mathbf{v} \in S.$$

In other words, every point on the line segment connecting $\mathbf{u}$ *and* $\mathbf{v}$ *is in* S*.*

Definition 2.1.17. *(**Convex Function**). Suppose that* f *is a real-valued function defined on a convex set* $S \in \mathbb{R}^d$ *. Then the function* f *is convex on* S *if*

$$f(\lambda\mathbf{u} + [1 - \lambda]\mathbf{v}) \leq \lambda f(\mathbf{u}) + [1 - \lambda] f(\mathbf{v}) \tag{2.1}$$

for all $\mathbf{u}, \mathbf{v} \in S$ *and all* λ *with* $\lambda \in [0,1]$. *The function* f *is strictly convex if the inequality is always strict for* $\mathbf{u} \neq \mathbf{v}$.

Example 2.1.18. *Examples of convex functions are*

- *exponential:* e^{ax}, *for any* $a \in \mathbb{R}$ *on domain* $\mathbb{R}$.
- *affine function:* $f(\mathbf{x}) = A\mathbf{x} + b$ *where* $f : \mathbb{R}^2 \to \mathbb{R}$.
- *the linear combination function* $h = \alpha f + \beta g$ *for* $\alpha, \beta \geq 0$, *given that* f *and* g *are convex.*

2.2 Calculus of Variation

Calculus of variations is a field of mathematical analysis that is concerned with the problem of extremising a functional (a function which depends on one or more functions) using variations, which are small changes in functions and functionals. This problem is a generalisation of the problem of finding extrema of functions of several variables. Details about this field can be found for example in [57, 61, 60].

Functionals are often expressed as definite integrals involving functions and their derivatives. Functions that maximise or minimise functionals may be found using the Euler–Lagrange equation of the calculus of variations. Consider a general functional $J(u) : \Omega \to \mathbb{R}$

$$J(u) = \int_{\Omega} F(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}))\, d\mathbf{x} \tag{2.2}$$

where Ω represents some normed vector space (for example, $\Omega = \mathbb{R}^d$, $d \geq 1$) that is a solution space for the unknown function $u(\mathbf{x})$. Here, the functional J depends upon the independent variable $\mathbf{x} = (x_1, x_2, .., x_d)$, $\nabla u(\mathbf{x}) = (\partial u/\partial x_1, \partial u/\partial x_2, ..., \partial u/\partial x_d)$ denotes the gradient of u, and $d\mathbf{x}$ is the $n-$differential element defined as $dx = dx_1 dx_2 ... dx_d$. We are concerned with the problem of minimising the functional $J(u)$ using calculus of variation:

$$\min_{u} J(u). \tag{2.3}$$

The necessary condition to be satisfied by any minimiser of a variational integral is the vanishing of its first variation δJ defined as:

$$\delta J(u) = \frac{d}{d\varepsilon} J(u + \varepsilon\varphi)\bigg|_{\varepsilon=0} = 0 \tag{2.4}$$

Here $\varphi \in \Omega$ represents a test function and ε is a real parameter. If u is a minimiser of $J(u)$, then (2.4) must be satisfied for all φ. We call $\delta J(u)$ the first variation of J at u in the direction of φ.

Definition 2.2.1. ***(Gateaux Derivative)****. Suppose* X *and* Y *are Banach spaces,* $U \subset X$ *is open, and* $J : U \to Y$. *The Gateaux differential derivative (or directional derivative or first variation) of* J *at* $u \in U$ *in the direction of* $\varphi \in X$ *is defined as*

$$\delta J(u:\varphi) = \frac{d}{d\varepsilon} J(u+\varepsilon\varphi)\Big|_{\varepsilon=0} = \lim_{\varepsilon\to 0} \frac{J(u+\varepsilon\varphi) - J(u)}{\varepsilon}$$

If the limit exists for all $\varphi \in X$, then $J(u)$ is Gateaux differentiable at u.

Definition 2.2.2. *(**Stationary Point**). Let $J : U \to \mathbb{R}$ be a functional with solution space $U \subset X$. For some $\tilde{u} \in U$, suppose J is Gateaux differentiable for all test functions $\varphi \in X$. Then $\tilde{u} \in U$is said to be a stationary point of J if $\delta J(\tilde{u}) = 0$ for all $\varphi \in X$.*

Definition 2.2.3. *(**Local Minimiser**). A functional $J : U \to \mathbb{R}$ is said to have a local minimiser at the point $\tilde{u}$ if there exists some $\varepsilon > 0$ such that*

$$J(\tilde{u}) \leq J(u), \forall u \in B_\varepsilon(\tilde{u}),$$

with $B_\varepsilon(\tilde{u}) := \{u \in U : \|u - \tilde{u}\| < \varepsilon\}$.

Definition 2.2.4. *(**Global Minimiser**). A functional $J : U \to \mathbb{R}$ is said to have a global minimiser at the point $\tilde{u}$ if*

$$J(\tilde{u}) \leq J(u), \forall u \in U.$$

The equation $\delta J(u) = 0$ is called the Euler-Lagrange equation for the minimisation problem in (2.3). If $J(u)$ is a convex functional, and U is a convex set, then every local minimiser of $J(u)$ is also a global minimiser.

In order to finalise this short section, we present an example of how to compute the first variation of a functional of our interest.

Example 2.2.5. *Consider the problem of finding the first variation of the functional*

$$J(u) = \int_\Omega |\nabla u| \, dxdy$$

defined on a domain $\Omega \subset \mathbb{R}^2$. We first introduce the small variation $\varepsilon\varphi$ composed of the parameter $\varepsilon \to 0$ and the continuously differentiable function φ in Ω . Then we compute,

$$\begin{aligned}
\tfrac{d}{d\varepsilon} J(u+\varepsilon\varphi)\big|_{\varepsilon=0} &= \frac{d}{d\varepsilon} \int_\Omega |\nabla(u+\varepsilon\varphi)| \, dxdy \Big|_{\varepsilon=0} \\
&= \int_\Omega \frac{\nabla(u+\varepsilon\varphi)}{|\nabla(u+\varepsilon\varphi)|} \cdot \nabla\varphi \, dxdy \Big|_{\varepsilon=0} \\
&= \int_\Omega \frac{\nabla u}{|\nabla u|} \cdot \nabla\varphi \, dxdy.
\end{aligned}$$

Now taking $v = \varphi$ and $\omega = \nabla u / |\nabla u|$ in Green's first identity, i.e.

$$\int_\Omega (\mathbf{w} \cdot \nabla v + v \nabla \cdot \mathbf{w}) \, d\mathbf{x} = \int_{\partial\Omega} v\mathbf{w} \cdot \mathbf{n} \, d\mathbf{s}$$

where $\partial\Omega$ is the boundary of Ω , $d\mathbf{s}$ indicates integration with respect to surface area on $\partial\Omega$, and $\mathbf{n}$ is the unit outward normal vector for each point $\mathbf{x} \in \Omega$. We obtain

$$\int_\Omega \nabla v \cdot \mathbf{w}\, dxdy = \int_\Omega \frac{\nabla u}{|\nabla u|} \cdot \nabla \varphi \, dxdy$$
$$= \int_{\partial\Omega} \varphi \frac{\nabla u}{|\nabla u|} \cdot \mathbf{n}\, ds - \int_\Omega \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \varphi \, dxdy$$

we further require

$$\frac{d}{d\varepsilon} J\left(u + \varepsilon\varphi\right)\Big|_{\varepsilon=0} = 0,$$

for all test function φ then the following partial differential equation known as the Euler Lagrange equation must be satisfied:

$$\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = 0 \;\; in \;\; \Omega$$

With Neumann boundary condition $\nabla u \cdot \mathbf{n} = 0$ on $\partial\Omega$.

2.3 Functions of Bounded Variation

In this section, we define the space of function of bounded variation (BV) which is an important type of function to many variational methods in imaging. Details can be found in [55, 56, 62, 80].

Let Ω be a bounded open subset of $\mathbb{R}^d$ and let $u \in L^1(\Omega)$. Define

$$\int_\Omega |Du| \; d\mathbf{x} = \sup_V \left\{ \int_\Omega u\left(\mathbf{x}\right) \nabla \cdot \varphi \, d\mathbf{x} \right\}$$

where V is the set of test functions

$$V = \left\{ \varphi = (\varphi_1, \varphi_2, ..., \varphi_d) \in C_0^1 \left(\Omega : \mathbb{R}^d \right) : \|\varphi(\mathbf{x})\|_{L^\infty(\Omega)} \leq 1, \forall \mathbf{x} \in \Omega \right\}$$

the divergence

$$\nabla \cdot \varphi = \sum_{i=1}^{d} \frac{\partial \varphi_i}{\partial x_i},$$

and C_0^1 is the space of continuously differentiable functions with compact support in Ω. As describe in [62] for a particular case $u \in C^1(\Omega; \mathbb{R}^d)$, integration by parts gives

$$\int_\Omega u \nabla \cdot \varphi \, d\mathbf{x} = - \int_\Omega \sum_{i=1}^{d} \frac{\partial u}{\partial x_i} \varphi_i \, d\mathbf{x}$$

for every $\varphi \in C_0^1(\Omega; \mathbb{R}^d)$, so that

$$\int_\Omega |Du| \; d\mathbf{x} = \int_\Omega |\nabla u| \; d\mathbf{x}$$

Definition 2.3.1. *(**Function of Bounded Variation**). A function $u \in L^1(\Omega)$ is said to have bounded variation in Ω if $\int_\Omega |Du| < \infty$. The notation $BV(\Omega)$ denotes the space of all functions in $L^1(\Omega)$ with bounded variation.*

A basic property of BV functions is the coarea formula which provides a connection between the total variation of a function and the perimeter of its level sets. For $u \in BV(\Omega)$ defined in Ω we define the level set in $\mathbb{R}^d$ as

$$E_\gamma = \{\mathbf{x} \in \Omega : u(\mathbf{x}) \leq \gamma\} \tag{2.5}$$

Definition 2.3.2. *(**Perimeter**). The perimeter of $E_\gamma \in \Omega$ is defined as*

$$Per(E_\gamma) = \int_\Omega \left|D\chi_{E_\gamma}\right| = \sup_V \left\{ \int_{E_\gamma} \nabla \cdot \varphi \, d\mathbf{x} \right\}$$

where χ_E is a characteristic or indicator function of the set E , defined as

$$\chi_E = \begin{cases} 1 & if\ x \in E \\ 0 & if\ x \in \Omega \backslash E. \end{cases}$$

Definition 2.3.3. *(**Coarea formula**). Let $u = u(\mathbf{x})$ and $f = f(\mathbf{x})$ be two scalar functions defined on $\mathbb{R}^d$. Assume that u is Lipschitz continuous and that for almost every $\lambda \in \mathbb{R}$, the level set $L = \{\mathbf{x} \in \mathbb{R}^d : u(\mathbf{x}) = \lambda\}$ is a smooth $(n-1)$-dimensional hypersurface in $\mathbb{R}^d$. Suppose also that f is continuous and integrable. Then*

$$\int_{\mathbb{R}^d} |\nabla u| \, f \, d\mathbf{x} = \int_{-\infty}^{\infty} \left(\int_L f \, ds \right) d\lambda.$$

For a particular case when $f = 1$ and the region of integration is a subset$\Omega \subset \mathbb{R}^d$ we get

$$\int_\Omega |\nabla u| \, d\mathbf{x} = \int_{-\infty}^{\infty} \left(\int_L ds \right) d\lambda = \int_{-\infty}^{\infty} Per(L, \Omega) \, d\lambda.$$

This shows that the total variation of a given function u is just the sum of every length of all its λ-level sets.

2.4 Inverse Problem and Tikhonov Regularisation

An inverse problem is one where the target is to recover the model parameter from some unknown observed data from a physical system. Mathematically, the inverse problem can be defined as below.

Definition 2.4.1. *(Forward and Inverse Problem). A forward (direct) problem is the process of computing the data y from the parameter x using a measurement operator f. The operator f maps the parameter in a function space X to the space of data Y. We denote*

$$y = f(x), \ for\ x \in X\ and\ y \in Y \tag{2.6}$$

as the connection between the parameter x and the data y. An inverse problem is the process to find the parameter $x \in X$ from the information of the data $y \in Y$ such that (2.6) *or an approximation of* (2.6) *holds.*

A common consideration with the inverse problems is that they are often ill-posed. According to the famous French mathematician, Hadamard [68] a problem is ill-posed if one of the following conditions does not hold

- a solution exists
- the solution is unique
- the solution depends continuously on the data (i.e. a small change in the data does not lead to a large change in the solution

and the problem is classified as well-posed if all conditions are satisfied.

Many problems in real life applications are ill-posed inverse problems. For example, given two data sets P and Q , we may need to compute the value of $R = P + Q$ which is an example of a forward problem. The problem becomes an inverse problem if we are asked to calculate the values of $P$and Q given the value of R. Ill-posed problems usually result from a lack of precise mathematical formulation and typically violate the stability condition where small changes in the given data lead to large changes in the solution [71]. To illustrate this, consider the following example.

Example 2.4.2. *Given*

$$A = \begin{bmatrix} 3 & 4 \\ 3 & 4+\varepsilon \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 0 \end{bmatrix}.$$

The forward problem is to compute $K = K(y) = Ay$, for $\varepsilon = 0$, which has solution $K = [9, 9]^T$. Meanwhile, the associated inverse problem is to compute y given this K. However for $\varepsilon = 0$, A is not invertible. As a result, there is no solution to the problem. If we change $\varepsilon = 10^{-6}$, then A becomes invertible and has a unique solution for $y = [3, 0]^T$. Perturbing K slightly to $\hat{K} = [9, 9 - \varepsilon]^T$, the solution to the inverse problem is $\hat{y} = [13/3, -1]^T$ which is considerably different from y since $|y|_2 = 9$ and $|\hat{y}| = 178/9$.

In 1963, the Russian mathematician Andrei N. Tikhonov [135] introduced the foundations of the theory of ill-posed problem solutions and developed the concept known as regularisation which transform an ill-posed problem into a well-posed problem. Basically, a new constraint is introduced in the problem which enforces the solution to belong to a specific set of solution.

Consider a given $A : F(A) \subseteq X \to Y$ operator between X and Y such that $Au = b$. If the solution u does not satisfy the well-posed criteria, Tikhonov [135] proposed to solve the following minimisation problem:

$$\min_u \|Au - b\|_2^2 + \alpha \|Lu\|_2^2 . \tag{2.7}$$

Here, the first term is the data, fedility or fitting term, the second is the regularisation term where L is a regularisation operator and $\alpha > 0$ regularisation parameter which determines the trade-off between regularisation and data fitting.

2.5 Discretisation of Partial Differential Equations

A continuous model is often transferred to a discrete version through discretisation mainly because the equation involved cannot be solved analytically or because only discrete data is available. For image processing tasks, the domain $\Omega \subset \mathbb{R}^2$ is normally rectangular and the values of a given function are known at uniformly distributed points in the domain. Therefore, the finite difference method is frequently used to discretise the domain.

We proceed by considering a 2-D problem with domain $\Omega = (a, b) \times (c, d)$ for $a, b, c, d \in \mathbb{R}$ on which we impose a $(n_x + 1) \times (n_y + 1)$ Cartesian grid with spacing $h_x = (b - a)/n_x$ and $h_y = (d - c)/n_y$ for the x and y direction respectively. We may consider vertex-centred discretisation where points are placed on the vertices of the grid mesh giving $(n_x + 1) \times (n_y + 1)$ grid points (x_i, y_j) located at

$$(x_i, y_j) = (a + ih_x, c + jh_y), \ for\ 0 \leq i \leq n_x \ and\ 0 \leq j \leq n_y.$$

In a cell-centred discretisation, grid points are placed at the centre of the grid cells giving $n_x \times n_y$ grid points (x_i, y_j) located at

$$(x_i, y_j) = \left(a + \frac{2i-1}{2}h_x, c + \frac{2j-1}{2}h_y\right), \text{ for } 1 \leq i \leq n_x \text{ and } 1 \leq j \leq n_y.$$

Operators such as derivatives in the PDE can be approximated using the Taylor expansions

$$u(x + h, y) = \sum_{i=0}^{\infty} \frac{h^i}{i!} \frac{\partial^i f(x, y)}{\partial x^i} \text{ and } u(x - h, y) = \sum_{i=0}^{\infty} (-1) \frac{h^i}{i!} \frac{\partial^i f(x, y)}{\partial x^i}.$$

Then, the approximation of the derivative $\partial f/\partial u$ at the point (i, j) called forward difference operator is given as

$$\nabla_+^x (u_{i,j}) \approx \frac{u(x + h) - u(x, y)}{h} = \frac{u_{i+1,j} - u_{i,j}}{h}.$$

The backward difference operator is defined as

$$\nabla_-^x (u_{i,j}) \approx \frac{u(x, y) - u(x - h, y)}{h} = \frac{u_{i,j} - u_{i-1,j}}{h}$$

and the central difference operator can be given as

$$\nabla_x^c (u_{i,j}) \approx \frac{u(x + h, y) - u(x - h, y)}{2h} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}$$

where $u_{i,j} = u(x_i, y_j)$ is the value of $u(x, y)$ at the point (i, j). The higher-order derivatives can be approximated in a similar way. For instance, the second-order approximation of $\partial^2 u/\partial x^2$ is defined as follows

$$\Delta^x(u_{i,j}) = \nabla^x_-\left(\nabla^x_+(u_{i,j})\right) = \frac{u_{i+1,j} - u_{i,j} + u_{i-1,j}}{h}$$

Similar definitions can be given for partial derivatives with respect to y.

2.6 Image Representation

In this section, we look at some of the most frequent ways of representing images. Computationally, a grey scale images is a 2-D array (matrix) $\mathbf{U} = [u_{i,j}]_{m\times n}$. Each entry of the array is called a pixel of the image and represents the level of brightness or intensity at that point. These values will be referred to as intensity values. The values typically either [0,255] or [0,1] depending upon if the image has been normalized or not, respectively. For colour or multi-channel images, the images are represented as multi-dimensional arrays $\mathbf{U} = [u_{i,j,k}]_{m\times n\times p}$ for an $m \times n$ pixels image. Here, p represents the number of channels of the image. For example, if an image $\mathbf{U}$ representing RGB colour scheme (Red, Green, Blue) than we have $\mathbf{U} = [u_{i,j,k}]_{m\times n\times 3}$. At an arbitrary pixel of the image, the intensity value is a vector $u_{i,j} = (u_{i,j,1}, u_{i,j,2}, u_{i,j,3})$ where each of the entries represent the intensity level at that pixel from each channels. In this thesis, however we just consider grey scale images for the experimental works in coming chapters.

To represent a grey scaled image mathematically, we characterised the image by a smooth function $u = u(x, y) : \Omega \to \mathbb{R}$ whose domain is given by a subset $\Omega \subset \mathbb{R}$. Based on this, images can be seen as the level sets or isophotes of the function u. For each real value λ define the λ-level set of u as

$$\gamma_\lambda = \{(x, y) \in \Omega : u(x, y) = \lambda\} \tag{2.8}$$

Then, the classical level set representation of u is the one-parameter family of all the level sets

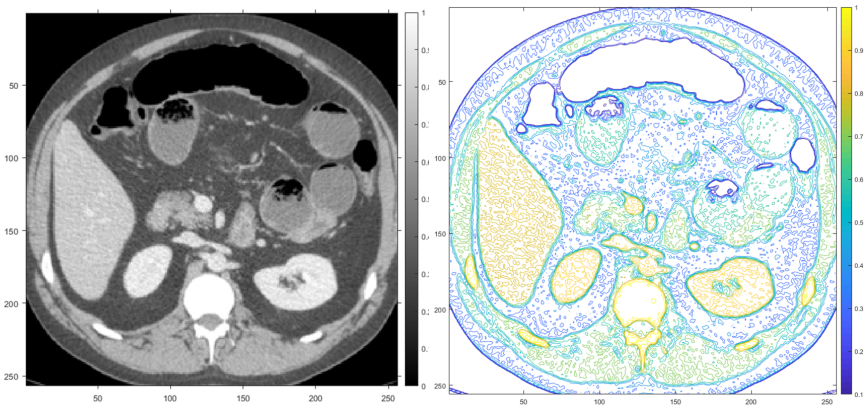
$$\Gamma_u = \{\gamma_\lambda : \lambda \in \mathbb{R}\}.$$

Figure 2.1 shows a grey scale image and some of its level set.

An image also can be represented as surface where the height is the grey scale value. The representation is obtained by considering an image as the induced 3-D surface or graph characterised by $z = u(x, y)$ which defines the image surface $(x, y, u(x, y))$. As the level set function $\phi(x, y, z) = u(x, y) - z$, we see that its zero level set $\{(x, y, z) : \phi(x, y, z) = 0\}$ corresponds to the surface $z = u(x, y)$. In Figure 2.2 , we show the surface representation of the image shown in Figure 2.1.

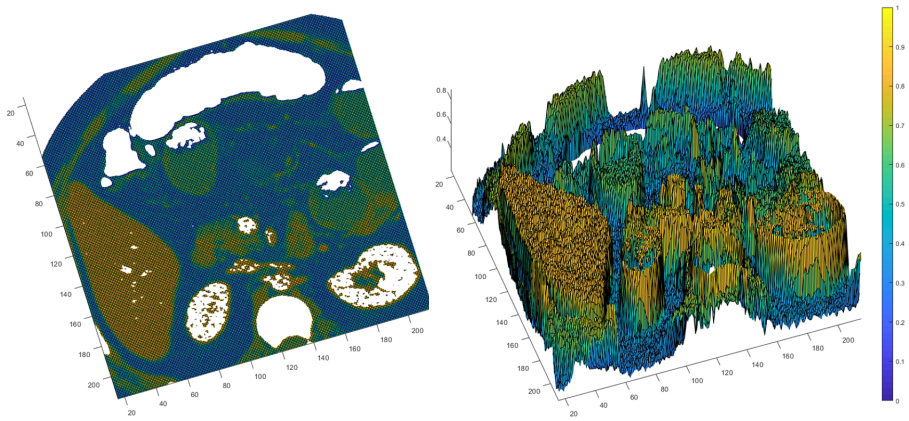
2.7 The Level Set Method

The level set method is a numerical technique devised by Osher and Sethian [107] for computing and analysing the motion of an interface in two or three dimensions. This method has been widely used in variety of problems including image segmentation,



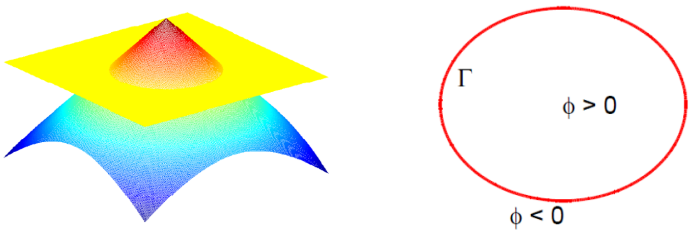
(a) (b)

Figure 2.1: (a) Original grey scale image of size 256×256 with intensity values on the interval [0,1]. (b) Level sets of the image in 2-D view.



(a) (b)

Figure 2.2: Two different views of the surface representation of the image in Figure 2.1



(a) (b)

Figure 2.3: Illustration of the interface Γ using the level set method: (a) shows a level set function $\phi(\mathbf{x})$ and its intersection with $\phi = 0$, (b) shows the corresponding interface, Γ implicitly represented by the zero level set of ϕ.

computational geometry, fluid dynamics flows, visualization, computer vision, control, visibility, restoration and many others [89, 123, 124, 125]. In this section we provide a brief overview of this method. We refer the reader to the original paper [107] and other work [123, 124, 125] for further details.

For a given interface (curve) $\Gamma \in \Omega$, the level set method amounts to implicitly represent Γ with the zero level set of a Lipschitz function $\phi : \Omega \to \mathbb{R}$ such that

$$\begin{cases} \phi(\mathbf{x}) > 0 & \text{inside } \Gamma \\ \phi(\mathbf{x}) < 0 & \text{outside } \Gamma \\ \phi(\mathbf{x}) = 0 & \text{on } \Gamma. \end{cases}$$

By defining the interface implicitly, topological changes of Γ are dealt with automatically. As the function ϕ evolves, the interface Γ can split or merge easily, as it is defined by the zero level set of ϕ. The interface for a corresponding level set function is illustrated in Figure 2.3.

As we are dealing with curve evolution, we will adjust the level set as a function of time t denoted by:

$$\begin{cases} \phi(\mathbf{x}(t), t) > 0 & \text{inside } \Gamma \\ \phi(\mathbf{x}(t), t) < 0 & \text{outside } \Gamma \\ \phi(\mathbf{x}(t), t) = 0 & \text{on } \Gamma. \end{cases}$$

Formally, we want to track the zero level set of ϕ that is $\phi(\mathbf{x}(t), t) = 0$. That means we need to derive the equation of motion of ϕ defined as

$$\frac{\partial \phi(\mathbf{x}(t), t)}{\partial t} = 0.$$

Using chain rule we obtain

$$\frac{\partial \phi}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}(t)}{\partial t} + \frac{\partial \phi}{\partial t} = 0$$

or in compact form,

$$\nabla \phi \mathbf{x}_t + \phi_t = 0.$$

The motion direction of the curve is normal, which is $\frac{\nabla\phi}{|\nabla\phi|}$ and there would also be a force, F that moves the curve. Hence, the speed is given by $\mathbf{x}_t = F\frac{\nabla\phi}{|\nabla\phi|}$. The previous motion equation can be written as:

$$\begin{aligned}\nabla\phi F\frac{\nabla\phi}{|\nabla\phi|} + \phi_t &= 0\\ F\,|\nabla\phi| + \phi_t &= 0\\ \phi_t &= -F\,|\nabla\phi|\end{aligned}$$

The initialisation is given as $\phi(\mathbf{x}, t=0) = d(\Gamma_0) = \phi_0$ where $d(\cdot)$ is generally a signed Euclidean distance function, whose zero level set is the initial contour Γ_0. Large variations in $\nabla\phi$ for general function F might increase the numerical computation error. Furthermore, as the interface evolves, ϕ generally drift away from its initialised level set and the level set function ϕ becomes too steep or too flat [131]. Therefore, it is common in practice to re-initialise the level set function as a signed distance function. This can be achieved by solving the following PDE for ϕ

$$\begin{aligned}&\phi_t - \operatorname{sgn}(\phi)\,(1 - |\nabla\phi|) = 0\\ &\phi\,(\mathbf{x}, 0) = \phi_0\end{aligned} \tag{2.9}$$

where ϕ_0 is the function which is supposed to be re-initialised. The procedure will convert the level set function to the unit distance function. We refer the reader to the work Sussman et al. [131] and the references therein, for further details.

2.8 Iterative Solutions to Equations

In this section, we discuss the methods to solve the problem that arise in later chapters. These methods are divided into three classes, which require different techniques. The first class is the basic iterative methods for solving linear equations. Here, we introduce the Jacobi method, followed by the Gauss-Seidel method and Kaczmarz method. The second class is iterative methods for nonlinear equations where we discuss Newton's method, gradient descent method, and additive operator splitting (AOS) method. Finally, the third class is multiresolution methods where the methods under discussion are multigrid method and multilevel method.

2.8.1 Basic Iterative Methods for Linear Systems

We introduce two well-known iterative methods for finding solutions to linear system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{2.10}$$

where A is $n \times n$ matrix, b is an $n \times 1$ vector and $\mathbf{x}$ is the $n \times 1$ vector of unknowns such that

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

For a square system of equations that is when the number of equations m is the same as the number of variables n , it is possible to have a solution. Hence, the following we will consider $m = n$. If $\mathbf{A}^{-1}$ exist, direct methods such as Gaussian Elimination can be used however they can be computationally very expensive especially when dealing with real application problems which require a lot of memory. Iterative methods such as Jacobi, Gauss-Seidel, and Kaczmarz can be very useful for solving a general linear system in (2.10) in terms of implementation and are computationally cheap. The process starts with an initial approximation $\mathbf{x}^{(0)}$ that generates a sequence of vectors $\left\{\mathbf{x}^{(k)}\right\}_{k=0}^{\infty}$ which gradually approximates the true solution $\mathbf{x}$ of the linear system (2.10). Such methods involve iterations of the form

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}, \tag{2.11}$$

for each iteration step $k = 1, 2, 3...$. In this way the system $\mathbf{Ax} = \mathbf{b}$ is converted into an equivalent system in (2.11) for some fixed matrix $\mathbf{T}$ and a vector $\mathbf{c}$, neither of which is dependent on the iterative step k. How $\mathbf{T}$ and $\mathbf{c}$ are defined depends on the technique used which we will now address.

The Jacobi Method

The Jacobi method is an iterative method which is simple to implement and often forms the basis for other methods. Given a system of liner equations (2.10), the i th equation is given by

$$\sum_{j=1}^{n} a_{i,j}x_j = b_i. \tag{2.12}$$

Solving (2.12) now for x_i, we get the equation

$$x_i = \frac{b_i}{a_{i,i}} + \sum_{j=1, j\neq i}^{n} \frac{-a_{i,j}x_j}{a_{i,i}}. \tag{2.13}$$

Then given all the components of $\mathbf{x}^{(k-1)}$ for $k \geq 1$, $x_i^{(k)}$ is generalised by

$$x_i^{(k)} = \frac{1}{a_{i,i}} \left(b_i + \sum_{j=1, j\neq i}^{n} -a_{i,j}x_j^{(k-1)} \right), \qquad \text{for } i = 1, 2, ..., n. \tag{2.14}$$

This method also can be implemented using parallel computation for speed gain. We may note that the computation of $x_i^{(k+1)}$ requires the values of each element of $\mathbf{x}^{(k)}$ except itself whereas the Gauss-Seidel method requires the entry $x_i^{(k)}$ with $x_i^{(k+1)}$ as

discussed below.

The Gauss-Seidel Method

Unlike the Jacobi method, Gauss-Seidel method uses the values of the current step rather than the previous step, i.e. to calculate $x_i^{(k)}$, Gauss-Seidel uses $x_i^{(k)}, ..., x_{i-1}^{(k)}$ while Jacobi method uses $x_i^{(k-1)}, ..., x_{i-1}^{(k-1)}$. This fact explains why the Gauss-Seidel method yields a better approximation than Jacobi method. In this way, Gauss-Seidel iterations can be written as

$$x_i^{(k)} = \frac{1}{a_{i,i}} \left(b_i + \sum_{j=1}^{i-1} -a_{i,j} x_j^{(k)} + \sum_{j=i+1}^{n} -a_{i,j} x_j^{(k-1)} \right), \qquad \text{for } i = 1, 2, ..., n. \tag{2.15}$$

From (2.15) we can easily notice that each new entry $x_i^{(k)}$ in the Gauss-Seidel method is very dependent on previously updated entries $x_j^{(k)}$ for all $j < i$, meaning the ordering of the equations is vital. As with Jacobi, this method also can be implemented in parallel to speed up computation.

The Kaczmarz Method

The Kaczmarz method is a technique that solve the problem one row of $\mathbf{A}$ at a time. In 2-D, each row of $\mathbf{A}$ can be thought of as defining a line given by $\mathbf{Ax}$ while in 3-D $\mathbf{Ax}$ is a plane, and above that it is called hyperplane. Given an initial guess $\mathbf{x}_0 = 0$, Kaczmarz's algorithm steps through each row of $\mathbf{A}$ moving to the point on the $\mathbf{A}_i\mathbf{x}$ hyperplane closet to the current estimate of $\mathbf{x}$. If $\mathbf{Ax} = \mathbf{b}$ has a unique solution, this approach will converge.

For the implementation, consider the hyperplane defined by $\mathbf{A}_{i+1}\mathbf{x} = \mathbf{b}_{i+1}$. Because the vector $\mathbf{A}_{i+1}^T$ is perpendicular to this hyperplane, the update to $\mathbf{x}^{(i)}$ from the constraint due to row $i + 1$ of $\mathbf{A}$ will be proportional to $\mathbf{A}_{i+1}^T$ and the update is given as

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha \mathbf{A}_{i+1}^T. \tag{2.16}$$

Using the fact that $\mathbf{A}_{i+1}\mathbf{x}^{(i+1)} = \mathbf{b}_{i+1}$ to solve for α, we obtain

$$\begin{aligned} &\mathbf{A}_{i+1}\left(\mathbf{x}^{(i)} + \alpha \mathbf{A}_{i+1}^T\right) = \mathbf{b}_{i+1} \\ &\mathbf{A}_{i+1}\mathbf{x}^{(i)} - \mathbf{b}_{i+1} = -\alpha \mathbf{A}_{i+1}\mathbf{A}_{i+1}^T \\ &\alpha = -\frac{\mathbf{A}_{i+1}\mathbf{x}^{(i)} - \mathbf{b}_{i+1}}{\mathbf{A}_{i+1}\mathbf{A}_{i+1}^T}. \end{aligned}$$

Thus the update formula is

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \frac{\mathbf{A}_{i+1}\mathbf{x}^{(i)} - \mathbf{b}_{i+1}}{\mathbf{A}_{i+1}\mathbf{A}_{i+1}^T} \mathbf{A}_{i+1}^T.$$

2.8.2 Iterative methods for Nonlinear Equations

We now consider the problem of finding solutions to systems of nonlinear equations. Such equations arise in many variational imaging problems. Let say that we want to solve the following nonlinear system

$$\begin{cases} F_1(x_1, x_2, ..., x_n) &= 0 \\ F_2(x_1, x_2, ..., x_n) &= 0 \\ \vdots &= 0 \\ F_n(x_1, x_2, ..., x_n) &= 0, \end{cases} \tag{2.17}$$

which may arise from discretisation of a nonlinear PDE or nonlinear optimisation problem

$$\min J(x_1, x_2, ..., x_n) \tag{2.18}$$

We can represent the system as $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ where

$$\mathbf{F} = (F_1, F_2, ..., F_n)^T, \quad \mathbf{x} = (x_1, x_2, ..., x_n)^T, \tag{2.19}$$

and $F_i : D \subset \mathbb{R}^d \to \mathbb{R},\ i = 1, ..., n$ are nonlinear operators which are continuously differentiable on $\mathbb{R}^d$. We need to find $\mathbf{x}^* \in \mathbb{R}^d$, a solution to the equation (2.17). We begin by introducing the Newton method, before discussing the gradient descent method and the AOS method.

The Newton Method

Let B denote the Jacobian matrix of $\mathbf{F}$

$$B_{i,j} = \frac{\partial F_i(\mathbf{x})}{\partial x_j}$$

and assume that B is Lipschitz continuous. Given the initial approximation $\mathbf{x}^{(0)}$, the method attempts to evaluate $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ using the following recurrence relation

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \left(B\left(\mathbf{x}^{(k-1)}\right)\right)^{-1} \mathbf{F}\left(\mathbf{x}^{(k-1)}\right). \tag{2.20}$$

The aim here is to find successively closer approximation to the solution $\mathbf{x}^*$. Computing the inverse of the Jacobian can however be a difficult task and can be avoided by rewriting (2.20) in a linear form as

$$B\left(\mathbf{x}^{(k-1)}\right) \mathbf{r}^{(k-1)} = -\mathbf{F}\left(\mathbf{x}^{(k-1)}\right).$$

for the unknown $\mathbf{r}^{(k-1)}$ and then setting $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{r}^{(k-1)}$. Assuming that the initial estimate is sufficiently close to the true solution, Newton's method can offer fast convergence.

The Gradient Descent Method

Similar to Newton's method, the descent method requires an initial approximation $\mathbf{x}^{(0)} \in \mathbb{R}^d$ of the solution and then iteratively approximate the solution using the scheme

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \alpha^{(k-1)}\mathbf{s}^{(k-1)}, \quad k = 1, 2, 3...$$

where the positive scalar $\alpha^{(k-1)}$ is called the step length which is not fixed and may take different values at each iteration, and $\mathbf{s}^{(k-1)}$ is a pre-defined search direction which we use to find the new iterate $\mathbf{x}^{(k)}$. If the search direction is given as the opposite direction to the gradient $\nabla F(\mathbf{x}^{(k-1)})$, the descent method is called as Steepest Descent (or Gradient Descent) method since the function F decreases fastest in this direction. The iterative scheme of the gradient descent method is given as

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \alpha^{(k-1)}\nabla F\left(\mathbf{x}^{(k-1)}\right), \quad k = 1, 2, 3...$$

For the descent type methods, it is expected that the function value is decreased with each iteration, that is

$$F\left(\mathbf{x}^{(k)}\right) \leq F\left(\mathbf{x}^{(k-1)}\right).$$

If we fix the step length to be equal to some time step Δt, then we obtain the Time Marching method. The iteration begins at time $t = 0$ and proceed in time until a solution is obtained. The time step must be chosen sufficiently small so that the method remains stable at each iteration which consequently increase the number of iterations required for convergence to a steady state solution, and hence $\nabla F = 0$. The time marching scheme is given as follows

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - t\nabla F\left(\mathbf{x}^{(k-1)}\right), \quad k = 1, 2, 3...$$

Despite its limitations in computational performance, its reliability and ease of implementation has made the time marching method very popular in solving many variational problem in image processing such as [34, 115].

The Additive Operator Splitting Method

While the explicit scheme such as time marching require very small time steps which leads to poor efficiency, the Additive Operator splitting (AOS) method introduced by [85, 140] is stable for larger time steps that result in improve computation time. The scheme applies to the diffusion equation in the following form

$$u_t\left(t, \mathbf{x}\right) = f\left(u\left(t, \mathbf{x}\right)\right) + \nabla \cdot \left(g \nabla u\left(t, \mathbf{x}\right)\right) \tag{2.21}$$

with initial and boundary condition

$$u\left(0, \mathbf{x}\right) = u^{(0)}\left(\mathbf{x}\right) \quad \text{and} \quad \frac{\partial u}{\partial \mathbf{n}} = 0 \quad \text{on } \Omega$$

where g is the diffusivity function while f denotes a reaction term. Letting $\mathbf{x} = (x_1, ..., x_n)$, we can rewrite (2.21) as

$$u_t = f + \sum_{i=1}^{n} \left(g u_{x_i}\right)_{x_i} = f + \left(g u_{x_1}\right)_{x_1} + ... + \left(g u_{x_n}\right)_{x_n}$$

In 1-D, the equation we consider is given as

$$u_t = f\left(u\right) + \left(g u_x\right)_x$$

After discretisation with respect to time, we get

$$\frac{u^{k+1} - u^k}{\Delta t} = f\left(u^k\right) + \left(g u_x^{k+1}\right)_x.$$

Further discretisation with respect to space yields

$$\begin{aligned}
&\tfrac{u^{k+1}-u^k}{\Delta t} \\
&= f\left(u_i^k\right) + \nabla_x \left(g_i^k \nabla_x \left(u_i^{k+1}\right)\right) \\
&= f\left(u_i^k\right) + \nabla_x \left(g_i^k \tfrac{1}{h} \left(u_{i+1/2}^{k+1} - u_{i-1/2}^{k+1}\right)\right) \\
&= f\left(u_i^k\right) + \tfrac{1}{h} \nabla_x \left(\tfrac{g_{i+1/2}^k + g_{i-1/2}^k}{2} \left(u_{i+1/2}^{k+1} - u_{i-1/2}^{k+1}\right)\right) \\
&= f\left(u_i^k\right) + \tfrac{1}{2h^2} \left(g_{i+1}^k + g_i^k\right) \left(u_{i+1}^{k+1} - u_i^{k+1}\right) - \tfrac{1}{2h^2} \left(g_i^k + g_{i-1}^k\right) \left(u_i^{k+1} - u_{i-1}^{k+1}\right) \\
&= f\left(u_i^k\right) + \tfrac{1}{2h^2} \left(g_{i+1}^k + g_i^k\right) \left(u_{i+1}^{k+1} - u_i^{k+1}\right) + \tfrac{1}{2h^2} \left(g_{i-1}^k + g_i^k\right) \left(u_{i-1}^{k+1} - u_i^{k+1}\right) \\
&= f\left(u_i^k\right) + \tfrac{1}{2h^2} \sum_{j \in \Psi(i)} \left(g_j^k + g_i^k\right) \left(u_j^{k+1} - u_i^{k+1}\right)
\end{aligned} \tag{2.22}$$

where $\Psi(i) = \{i-1, i+1\}$ is the set of neighbours of i. Equation (2.22) can be rearranged as

$$u_i^{k+1} - \frac{\Delta t}{2h^2} \sum_{j \in \Psi(i)} \left(g_j^k + g_i^k\right) \left(u_j^{k+1} - u_i^{k+1}\right) = u_i^k + \Delta t f\left(u_i^k\right) \tag{2.23}$$

Since we have the following relation

$$\sum_{j \in \Psi(i)} \left(g_j^k + g_i^k\right) \left(u_j^{k+1} - u_i^{k+1}\right) = \left(\sum_{j \in \Psi(i)} \left(g_j^k + g_i^k\right) \left(u_j^{k+1}\right)\right) - u_i^{k+1} \sum_{j \in \Psi(i)} \left(g_j^k + g_i^k\right)$$

we can rewrite equation (2.23) in matrix form as

$$W^k u^{k+1} = u^k + \Delta t f^k, \qquad W^k = I - \frac{\Delta t}{2h^2} A^k \tag{2.24}$$

where the matrix $A^k = \left[a_{i,j}^k\right]$ is defined as

$$a_{i,j}^k = \begin{cases} -\sum_{j\in\Psi(i)} \left(g_j^k + g_i^k\right) & \text{if } j = 1 \\ g_j^k + g_i^k & \text{if } j \in \Psi(i) \\ 0 & \text{otherwise.} \end{cases}$$

Without loss of generality we drop the term f and this yields

$$u^{k+1} = \left(I - \frac{\Delta t}{2h^2} A^k\right)^{-1} u^k. \tag{2.25}$$

This tridiagonal system is solved using the Thomas Algorithm [141].

To demonstrate the reliability of the AOS, we will briefly discuss the established criteria for nonlinear diffusion scale-space and the advantages of satisfying such conditions [142, 140]. For a given discrete scheme of type

$$u^0 = f \tag{2.26}$$

$$u^{k+1} = Q(u^k)u^k, \quad \forall k \in \mathbb{N}_0 \tag{2.27}$$

the following criteria must hold:
(D1) Continuity in its argument:

$$Q \in C\left(\mathbb{R}^N, \mathbb{R}^{N\times N}\right)$$

(D2) Symmetry:

$$q_{ij} = q_{ji}, \quad \forall\, i, j \in J$$

(D3) Unit row sum:

$$\sum_{j\in J} q_{ij} = 1, \quad \forall\, i \in J$$

(D4) Nonnegativity:

$$q_{ij} \geq 0, \quad \forall\, i, j \in J$$

(D5) Positive diagonal:

$$q_{ii} \geq 0, \quad \forall\, i \in J$$

(D6) Irreducibility: For $\forall\, i, j \in J$ there exist $k_0, ..., k_r \in J$ with $k_0 = i$, and $k_r = j$ such that $q_{k_p k_{p+1}\neq 0}$ for $p = 0, ..., r-1$.

In Weickert et al. [141], the AOS scheme is demonstrated to have fulfilled the criteria for nonlinear diffusion scale-space above. This makes the scheme unconditionally stable and it will not suffer from any time step size restriction.

The 2-D diffusion equation can be written in the form

$$u_t\left(t,\mathbf{x}\right)=f\left(u\left(t,\mathbf{x}\right)\right)+\sum_{j=1}^{2}\left(g_j(u)u_{x_j}\left(t,\mathbf{x}\right)\right)_{x_j} \tag{2.28}$$

with initial condition $u(0,\mathbf{x})=u_0(\mathbf{x})$ and boundary condition $\partial u/\partial\mathbf{n}=0$ for $\mathbf{x}\in\partial\Omega$. To discretise, let Δt represent the step size. Then, the discrete time steps $t_k=k\Delta t,\ k\in\mathbb{N}$. We denote h_ι be the grid step size in the direction ι. Denoting $u_i^k=u(x_i,t_k)$ and $g_i^k=g(u(x_i,t_k))$. Then, the equation in discreet form is given as

$$\frac{u_i^{k+1}-u_i^k}{\Delta t}=\sum_{\iota=1}^{2}\sum_{j\in\Psi_\iota(i)}\frac{g_j^k+g_i^k}{2h_\iota^2}\left(u_j^{k+1}-u_i^{k+1}\right)+f\left(u_i^k\right).$$

Without loss of generality we drop the reaction term f and we have

$$u^{k+1}=\left(I-\Delta t\sum_{\iota=1}^{2}A_\iota(u)\right)^{-1}u^k.$$

The AOS scheme proposes

$$u^{k+1}=\frac{1}{2}\left(\sum_{\iota=1}^{2}\left(I-2\Delta tA_\iota\right)\right)^{-1}u^k,$$

solved efficiently using the Thomas algorithm [141].

Unlike the alternating implicit method (ADI) scheme, in which the subsystems must be solved sequentially, the split equations can be solved concurrently in the AOS scheme. In addition, the AOS method is well suited to be implemented in parallel processors [144].

2.8.3 Multigrid method and Multilevel Optimisation Method

The multigrid method and multilevel optimisation method are developed using the idea of hierarchy of discretisation where there is a pyramid of grids. The methods are examples of a class of techniques called multiresolution methods which aim to accelerate the convergence of a basic iterative method which efficient to handle problem with huge data. The multigrid method is based on optimise-discretise scheme where it solves a PDE that may arise from a variational problem numerically. In contrast, the multilevel optimisation method is based on discretise-optimise scheme where minimisation of a variational problem is solved directly without using PDE.

The Multigrid Method

This method was first introduced in the 1970s by Brandt [19]. It has two basic principles, the error smoothing and coarsening principles. The relaxation or iterative techniques discussed above have what is known as the smoothing effect on the error. These iterative techniques reducing rapidly the high frequency components of the error of the solution but may not be effective at reducing the low frequency component of the error. The

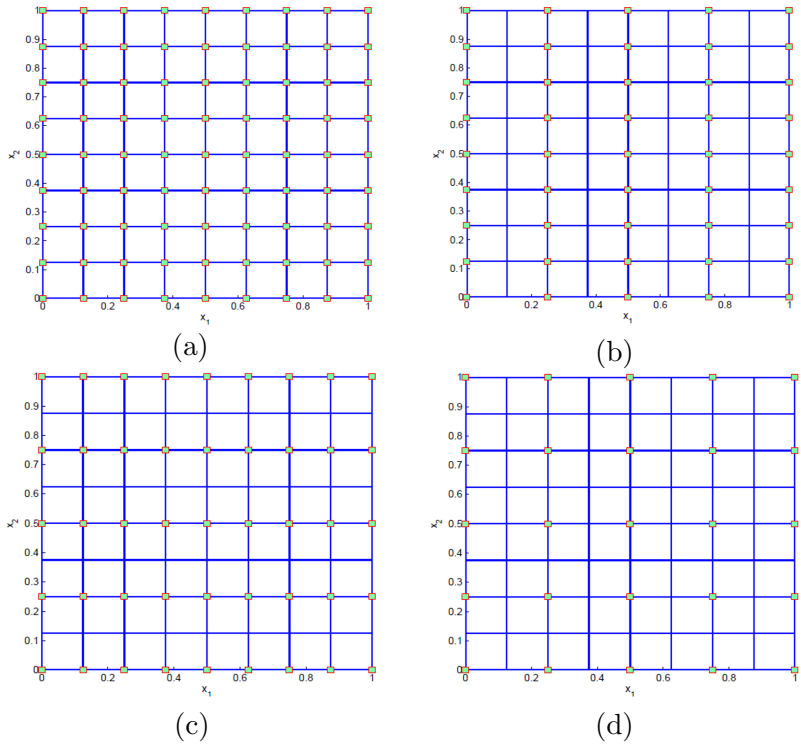


Figure 2.4: Illustration of the standard coarsening strategy. (a) represents the fine grid with 9×9 discretisation points. The coarse grid in (b) is obtained doubling the mesh size in the x_1-direction while in (c), the coarse grid is obtained by doubling the mesh size in the x_2-direction. Finally, the coarse grid (d) is constructed using these standard procedures.

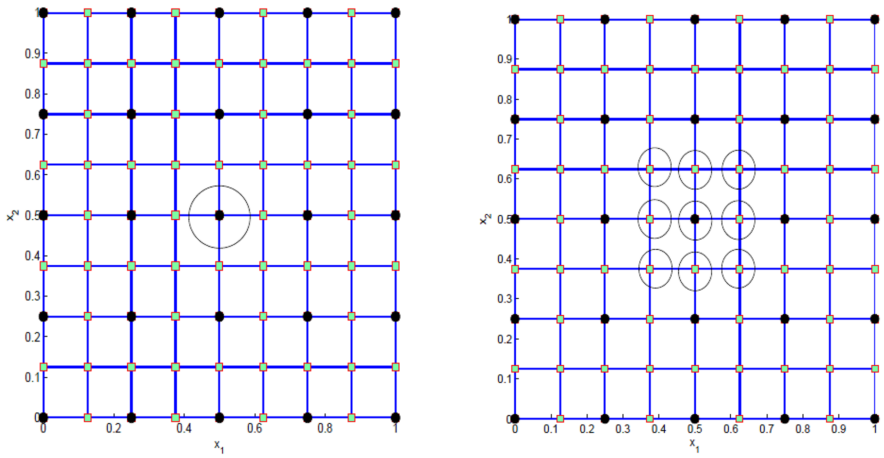
coarsening principle states that smooth error term have a good approximation on a coarse grid.

To illustrate precisely the main idea of multigrid methods, let us now focus on the linear system $A_h u_h = f_h$ resulting from an elliptic PDE on the fine grid Ω^h with grid spacing (h, k). Let v_h be an approximation solution computed by performing a few steps with a smoother on the fine grid problem This step is known as pre-smoothing step. Then, the residual or defect equation is defined as

$$A_h e_h = f_h - A_h v_h = r_h, \tag{2.29}$$

where $e_h = u_h - v_h$ is the error of the solution. We will now construct a course grid Ω^H with grid spacing (H, K) for the grid Ω^h. A typical standard coarsening is to double the spacing i.e $H = 2h,\ K = 2k$. If Ω^h has $(n+1) \times (m+1)$ grid points including boundary then $\Omega^{H=2h}$ will have$(n/2+1) \times (m/2+1)$ including the boundary points. The coarse grid will be a subset of the fine grid. Figure 2.4 illustrates standard coarsening for vertex-centred discretisation points [71].

Since the high frequency components of the error in pre-smoothing step have already been reduced by the smoother, we can transfer the following residual equation to the



(a) Injection operator (b) Full weighting operator

Figure 2.5: Illustration of restriction operators. (a) is the injection operator and (b) is the full weighting operator for vertex-centred discretisation. The black points are the coarse points. The points in circles are the active points used to compute the coarse points for each operator.

coarse grid as follows

$$A_h e_h = r_h \rightarrow A_H e_H = I_h^H r_h = r_H. \tag{2.30}$$

Here A_H is assumed to be an appropriate approximation of A_h on the coarse grid and I_h^H is the restriction operator i.e transfer operator from the fine to coarse grid defined as

$$\upsilon_{H=2h} = I_h^{H=2h} \upsilon_h. \tag{2.31}$$

The most obvious restriction operator is called injection operator defined as

$$\upsilon_{i,j}^{H=2h} = \upsilon_{2i,2j}^h \tag{2.32}$$

which is simple and fast, however not robust as it simply copies υ^{2h} from the values of υ^h at the same point on the fine grid. This ignores the odd-numbered fine grid values $\upsilon_{2i+1,2j+1}^h$. An alternative is a smoothing map, also known as a full weighting operator defines as

$$\begin{aligned} \upsilon_{i,j}^{H=2h} = \tfrac{1}{16}[&\upsilon_{2i-1,2j-1}^h + \upsilon_{2i-1,2j+1}^h + \upsilon_{2i+1,2j-1}^h + \upsilon_{2i+1,2j+1}^h + \\ &2\left(\upsilon_{2i,,2j-1}^h + \upsilon_{2i,2j+1}^h + \upsilon_{2i-1,2j}^h + \upsilon_{2i+1,2j}^h\right) + 4\upsilon_{2i,2j}^h]. \end{aligned} \tag{2.33}$$

We illustrate how both restriction operators are performed in Figure 2.5.

After the residual equation system (2.30) on the coarse grid have been solved exactly, the coarse grid correction e_H is then interpolated back to the fine grid e_h by the interpolation operator I_H^h i.e $e_h = I_H^h e_H$. The most commonly used interpolation or prolongation operator is the bilinear operator

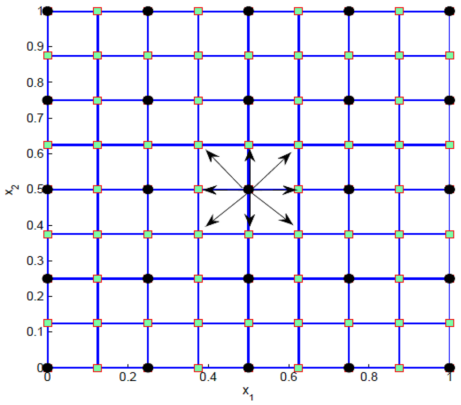


Figure 2.6: Illustration of bilinear operator from the coarse grid to the fine grid. The coarse point in black are used to obtain all the nine fine points surrounding it.

$$v_h = I^h_{H=2h} v_{H=2h}$$

where

$$\begin{aligned} v^h_{2i,2j} &= v^{2h}_{i,j} \\ v^h_{2i+1,2j} &= \frac{1}{2}\left(v^{2h}_{i,j} + v^{2h}_{i+1,j}\right) \\ v^h_{2i,2j+1} &= \frac{1}{2}\left(v^{2h}_{i,j} + v^{2h}_{i,j+1}\right) \\ v^h_{2i+1,2j+1} &= \frac{1}{4}\left(v^{2h}_{i,j} + v^{2h}_{i+1,j} + v^{2h}_{i,j+1} + v^{2h}_{i+1,j+1}\right). \end{aligned}$$

The coarse grid point that coincides with the fine grid point is left unchanged, and the surrounding fine grid points receive a contribution depending on the neighbourhood relation. In addition, this operator is the adjoint operator to the full weighting operator. We illustrate the bilinear interpolation operator in Figure 2.6.

Now, we can update the approximated solution v_h of the original linear system on the fine grid by $v_h^{new} = v_h + e_h$. This step is called coarse-grid correction step. The last step called post-smoothing step is to perform the smoother again to remove high frequency part of the interpolated error. This procedure is similarly applied to the case nonlinear PDE but with suitable nonlinear smoother like the ones discussed in Section 2.8.2.

The Multilevel Optimisation Method

In [33] Chan and Chen propose an alternative to multigrid PDE approach called multilevel optimisation method which uses local optimisation and correction via coarse grids. The method is applied to the discrete nondifferentiable functional of Total Variation image denoising optimisation problem. A brief outline of the method is given below.

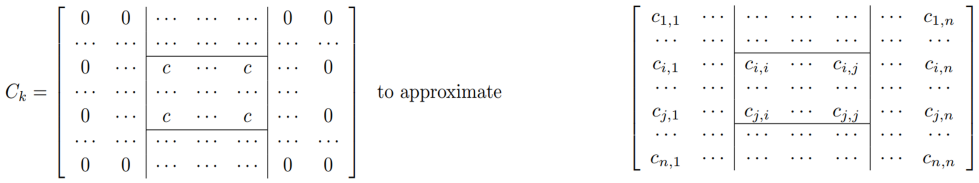


Figure 2.7: The illustration of $C_k = P_k\hat{c}$ reproduced from [33].

Given an optimisation problem

$$\min_{u\in\mathbb{R}^{n\times n}} f(u) \tag{2.34}$$

with the assumption of $n = 2^L$ and let the standard coarsening be used giving rise to $L+1$ levels: $k = 1\,(\text{finest}), 2, ..., L, L+1\,(\text{coarsest})$. The dimension of level k is denoted as $\tau_k \times \tau_k$ with $\tau_k = n/2^{k-1}$. Several steps of a local optimisation method are applied to the local optimisation problem $\min_{u_{i,j}} f_{i,j}(u_{i,j})$ which results from freezing all non (i,j) component of u at their current value. This step is either solved analytically (if possible) or via iteration to generate an approximation of problem (2.34).

Assume that $\tilde{u} \in \mathbb{R}^{n\times n}$ is the current approximation to (2.34). We wish to find the best piecewise constant function $C \in \mathbb{R}^{n\times n}$ so that it is the solution of the following:

$$\min_{c} f(\tilde{u} + c). \tag{2.35}$$

The minimisation problem (2.35) is equivalent to the original problem (2.34) and it is solved on coarse levels:

$$\hat{c} = \underset{c\in\mathbb{R}^{\tau_k\times\tau_k}}{\arg\min}\, f(\tilde{u} + P_k c), \qquad C_k = P_k\hat{c}. \tag{2.36}$$

Here, $P_k : \mathbb{R}^{\tau_k\times\tau_k} \to \mathbb{R}^{n\times n}$ is the interpolation operator, so $C_k \in \mathbb{R}^{n\times n}$. The illustration of $C_k = P_k\hat{c}$ is given in Figure 2.7.

Once $\hat{c}$ is obtained after few iterations, $\tilde{u}$ is updated by

$$u^{new} = \tilde{u} + P_k\hat{c}.$$

The method may get stuck to local minima due to non-differentiability of the energy functional. The wrong solution is associated with flat patches. To overcome that situation, Chan and Chen [33] have proposed the "patch detection" idea in the formulation of the multilevel method for image denoising problems. The patch detection idea searches the entire image for the possible patch size on the finest level after each multilevel cycle and an extra coarse grid based on these patches are added and the piecewise constant update is implemented as above.

Chapter 3

Review of Variational Models in Image Processing

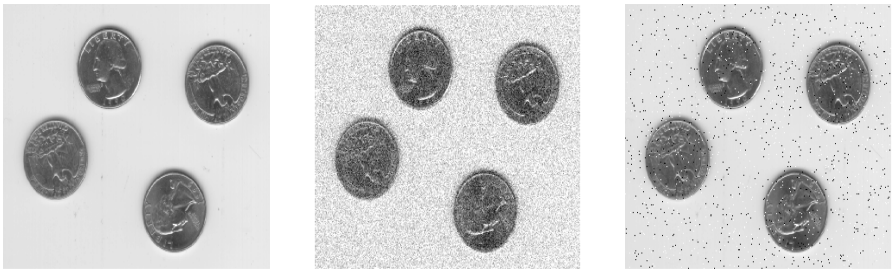
3.1 Introduction

In this chapter we briefly review variational models for image denoising and image segmentation, introducing relevant models of particular interest to our work in this thesis. We begin with introducing variational models in image denoising in Section 3.2. A seminal approach to this imaging problem was introduced in 1992 by Rudin, Osher, and Fatemi (ROF) [115]. This model that makes used the total variation (TV) regularisation is important to our work, as it is closely related to many segmentation models of our interest that also involve TV term. The segmentation models are discussed in Section 3.3.

3.2 Variational Models for Image Denoising

Image denoising is a research topic that has drawn much attention within the last decades for noise removal in signals and images. The noise in digital images is the common cause of degradation to an image normally produced during image acquisition such as using a scanner and digital camera [72]. The acquisition process which converts an optical image into continuous electrical signal that is then sampled is the primary source of noise [137]. Practically, it is almost impossible to remove noise totally without distorting an image. However, it is imperative that noise is reduced to a certain acceptable level for further analysis of the image.

There are two most common types of noise, Gaussian noise and salt-and-pepper noise. Gaussian noise is additive noise, occurs due to electronic noise in the image acquisition system, while the salt-and-pepper noise is may cause by malfunctioning pixel elements in the camera sensors. Faulty memory locations or timing errors in the digitisation process [137]. Figure 3.1 shows an image corrupted with Gaussian and salt-and-pepper noise. Further details about types of noise can be found in [31] and understanding this is vital to modelling suitable noise removal.



(a) (b) (c)

Figure 3.1: (a) Original grayscale image. (b) Gaussian noise corrupted image. (c) Salt-and-pepper noise corrupted image.

In variational framework, an edge preserving image denoising model with desirable mathematical properties was introduced by Rudin, Osher, and Fatemi (ROF) in their seminal work [115]. The model proposes to minimise the following functional

$$\min_u \left\{ \alpha \int_\Omega |\nabla u| \; dxdy + \frac{1}{2}(u-z)^2 \, dxdy \right\}, \tag{3.1}$$

Where the parameter $\alpha > 0$ represents a trade-off between the quality of the solution and the fit to the observed data $z(x, y)$. The first term is the regularisation term using the total variation of $u(x, y)$ while the second term is a fidelity or data term to ensure that the resulting denoised image $u(x, y)$ will be close to the given image $z(x, y)$. The formulation (3.1) is a well-posed problem so existence and uniqueness of its minimiser is guaranteed [27].

Minimisation of (3.1) is originally done using optimise-discretise scheme by solving the associate Euler Lagrange equation, derived using calculus of variation as discussed in previous chapter. The Euler Lagrange equation is given formally as follows

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + u - z = 0 \;\; in \;\; \Omega, \tag{3.2}$$

with Neumann boundary condition $\nabla u \cdot \mathbf{n} = 0 \;\; on \;\; \partial\Omega$.

Notice that for equation (3.2), singularity may exist when the nonlinear coefficient $|\nabla u| = 0$. A typical remedy is to approximate minimisation problem in (3.1) by

$$\min_u \left\{ \alpha \int_\Omega \sqrt{|\nabla u| + \beta} \, dxdy + \frac{1}{2}(u-z)^2 \, dxdy \right\}, \tag{3.3}$$

for small parameter $\beta > 0$. Hence, the associate Euler Lagrange for (3.3) is given as

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u| + \beta}} \right) + u - z = 0 \text{ in } \Omega,$$
$$\frac{\partial u}{\partial \mathbf{n}} = 0 \text{ on } \partial\Omega.$$

Another way to solve the problem in (3.1) is using discretise-optimise scheme as done in [33] where the optimisation problem is solved directly without approximation.

Of course, there exist many more solvers for the problem (3.1) in literature, for instance see [35, 116, 117, 139, 28, 63] and references therein. The main drawback with the TV model is that it transforms piecewise smooth function into piecewise constant functions. This phenomenon is known as staircase effect. This makes denoised images look blocky.

Many effort has been made (see for instance [42, 91, 117] and the references therein) to reduce the staircase effect in second order models numerically. Moreover, some researchers have turned to higher order models trying to reduce the staircase problem [86, 87, 149, 22].

3.3 Variational Models for Image Segmentation

Image segmentation is defined as the process of dividing a digital image into multiple regions (sets of pixels) of shared characteristics such as intensity, texture and colour, or in other words it is a process to distinguish objects from the background [6, 31, 98]. This field has broad applications, for example; in computer vision, fingerprints or face identification, computer graphic, astronomy and medical image processing. Image segmentation can be classified into two approaches: the non-equation (non-variational) approach and the variational or energy functional-based approach. The non-variational approach such as thresholding techniques, region merging algorithms, the watershed techniques and so on are considered as a simple way to segment an image. However, they are not defined in rigorous mathematical framework compared to variational approach, for more details see [127]. In variational approach, the minimiser of the energy functional corresponds to a meaningful representation of the image. The main target is to find a closed contour Γ that partitions a domain $\Omega \in \mathbb{R}^2$ into subregions $\Omega_s,\ s = 1, 2, ..., N$. In the following sections we introduce seminal works from the subject that related to our work in this thesis.

3.3.1 Mumford-Shah Approach

Mumford and Shah [102] introduced a way to partition a given image $z(x, y)$ by piecewise smooth function. Let Ω be a bounded domain in $\mathbb{R}^n$. The n-dimensional Mumford-Shah functional can be defined as

$$F^{MS}(u, \Gamma) = \alpha H^{n-1}(\Gamma) + \mu \int_{\Omega} (u - z)^2 \, dxdy + \int_{\Omega \setminus \Gamma} |\nabla u|^2 \, dxdy, \tag{3.4}$$

where α and μ are positive tuning parameter, H^{n-1} is the $(n-1)$-dimensional Hausdorff measure (that is, the usual $(n-1)$-dimensional Hausdorff measure is $(n-1)$-dimensional area in case of subsets of regular hyper surfaces, and the most relevant case $n = 2$ is the length). The functional consists a data term on $u \in C^1$, forcing u to approximate z and two regularity terms. The first regularity term imposes smoothness on u and the other imposes regularity on Γ which requires the boundaries to be as short and smooth as possible. Theoretical results on the existence and regularity of minimizers of (3.4) can be found in [102]. A reduced case of the above model is obtained by assuming u

as a piecewise constant function inside each connected region Ω_i that is $u = c_i$ which implies the average values of z in each region Ω_i. Thus, the reduced (piecewise constant) Mumford-Shah functional is given as

$$F^{MS2}(u, \Gamma) = \alpha |\Gamma| + \mu \sum_i \int_{\Omega_i} (z - c_i)^2 \, dxdy. \tag{3.5}$$

In practise, the functionals (3.4)and (3.5) are difficult to minimise because of the unknown set Γ and also the functionals are nonconvex. A possible solution of these problems will be addressed in Section 3.34.

3.3.2 Snake: Active Contour Model

Although the Mumford-Shah model (3.4) aims is to extract all significant parts in an image, some specific parts of the image can be more important than others depending on applications. For example, in medical imaging radiologist aims to analyse a particular organ or tumor in human body. The active contour model introduced by Kass et.al in [77] aims at detecting a particular object in an image based on image edge.

Let Ω be a bounded and open set in $\mathbb{R}^n$, with $\partial\Omega$ its boundary. Let z be a given image and denote $C(s) : [0, 1] \to \mathbb{R}^n$ a piecewise $C^1([0, 1])$ planar curve, represented as $C(s) = (x(s), y(s)) \in \Omega,\ s \in [0, 1]$. The snakes model amounts to minimise the following energy functional

$$F^{KWT}(C(s)) = \alpha \int_0^1 \left|\frac{\partial C}{\partial s}\right|^2 ds + \beta \int_0^1 \left|\frac{\partial^2 C}{\partial s^2}\right|^2 ds + \lambda \int_0^1 g(\nabla z(C))^2 ds. \tag{3.6}$$

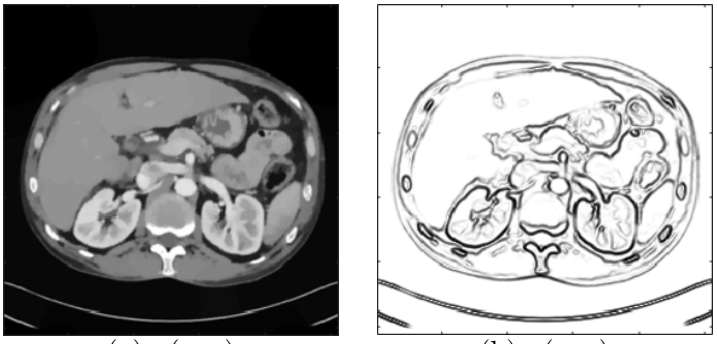
Here, α, β and λ are positive constant. The first term is the elasticity energy and the second term is bending energy. Both energy terms is called internal energy and responsible in determining the continuity and the smoothness of the curve. The third term is the external energy and attracts the contours towards the edge of the object in the image z. The function $g(\nabla z)^2$ is an edge detector given by

$$g(\nabla z) = \frac{1}{1 + \gamma |\nabla (z * G_\sigma)|^2}, \tag{3.7}$$

where γ is a positive constant. The function G_σ is the Gaussian smoothing function with standard deviation σ and mean μ such that $G_\sigma = \frac{1}{2\pi\sigma^2} \exp^{-\left|(x-\mu)^2 + (y-\mu)^2\right|/2\sigma^2}$ and $z * G_\sigma$ is a smooth version of z. For a given image z, its gradient i.e ∇z has high values in the neighbourhood of each objects in the image due to intensity change. An example of this edge function for a given image can be seen in Figure 3.2.

The equation (3.7) tells us that it takes a near zero value on the edge. Thus, the minimisation of functional (3.6) will push the contour towards the edge. The associate Euler Lagrange equation for (3.6) is given as

$$-\alpha \frac{\partial^2 C}{\partial s^2} + \beta \frac{\partial^4 C}{\partial s^4} + \lambda \nabla g^2 = 0. \tag{3.8}$$



(a) $z(x, y)$ (b) $g(x, y)$

Figure 3.2: (a) is a given image $z(x, y)$and (b) is the edge detection function, $g(x, y)$ defined in equation (3.7).

The solution for (3.8) using finite difference is not unique and strongly dependent on initialisation contour due to nonconvexity of functional (3.6) [77], hence it has local minima. On top of that, it does not allow changes of topology as the initial and final curve has the same topology, consequently this model is not possible to segment more than one object in an image.

To overcome the limitation of the topology change, the level set method [106, 107, 124] can be used. As we will show in the following section, the curve C is implicitly represented by a function of higher dimension ϕ, called the level set function. Therefore, the curve evolution equation can be rewritten in a level set formulation.

3.3.3 Geodesic Active Contours

V. Casselles et al. [25] proposed a new and improved model based on Kass et al. [77] called geodesic active contour model defined below

$$F^{GAC}\left(C(s)\right) = \int_0^{L(c)} g\left(\left|\nabla z\left(C(s)\right)\right|\right) ds, \tag{3.9}$$

where $L(C)$ is the Euclidean length of the curve C. The function g is the edge detecting function defined in (3.7). The Euler-Lagrange equation of the functional and gradient descent gives the following PDE

$$\frac{\partial C}{\partial t} = g\kappa\mathbf{n} - \left(\nabla g \cdot \mathbf{n}\right)\mathbf{n}, \tag{3.10}$$

where κ is the Euclidean curvature and $\mathbf{n}$ is the unit normal vector. The equation (3.10) shows how each point in the geodesic active contour should move in order to decrease the length F^{GAC}. The final solution that is the segmented object is then given by the steady state solution of (3.10). In level set formulation, the evolution equation is given as

$$\frac{\partial \phi}{\partial t} = |\nabla\phi|\left(\nabla \cdot \left(g\frac{\nabla\phi}{|\nabla\phi|}\right) + \nu g\right), \tag{3.11}$$

where ϕ is a Lipschitz function representing C as a zero level set. The term $\nu\, g$, $v > 0$ is added to increase the evolution speed and at the same time to ensure the curves is attracted towards the object boundary.

Despite the success of Geodesic Active Contours in handling topological changes in an image, it is limited in terms of applications in two senses. Firstly, the GAC model rely on the edge function g which depends on the gradient of the image. This means that images that either contains noise or have weak boundaries i.e not well defined boundaries are not suitable for this model. If that is the case, the final contour may segment the targeted object including the noise or may pass through the object with weak boundaries [38]. If the image is too noisy one may set the smoothing Gaussian in the edge detector function to be strong, however this can smooth the edges too. Secondly, the GAC model is nonconvex and is therefore highly sensitive to initialisation. In the next section, we introduced another active contour model which is independent of edge function to stop the contour at edges.

3.3.4 Chan-Vese Model

Chan and Vese (CV) model [38] proposed a new energy based model for image segmentation without the use of the image gradient as a stopping criterion, instead the stopping function depends on Mumford-Shah functional [102]. This model also known as known as Active Contour Without Edges [38] . The technique consider a special case of the piecewise constant Mumford-Shah functional [102] where the functional is restricted to only two phases, representing the foreground and the background of the image $z\,(x, y)$.

The assumption behind the model is that the observed image z is formed by two regions of approximately piecewise constant intensities of distinct (unknown) intensity values c_1 and c_2, separated by the curve or contour Γ. Let the object to be detected is represented by the region with the value c_1 inside the curve Γ whereas outside Γ, the intensity of z is approximated with the value c_2. Then, the Chan-Vese model minimises the following variational formulation as follows

$$\begin{aligned} F^{CV}\,(c_1, c_2, \Gamma) &= \mu \cdot \text{Length}\,(\Gamma) + \nu \cdot \text{Area}\,(\text{inside}\,(\Gamma)) \\ &+\lambda_1 \int_{\text{inside}(\Gamma)} (z - c_1)^2 dxdy + \lambda_2 \int_{\text{outside}(\Gamma)} (z - c_2)^2 dxdy \end{aligned} \tag{3.12}$$

Here, the fixed parameters are $\mu \geq 0$, $\nu \geq 0$, λ_1,$\lambda_2 > 0$. In general, the area constraint is ignored by letting $\nu = 0$ and the parameter for the fitting terms are evenly balanced i.e, $\lambda_1 = \lambda_2$. Then, the minimisation problem is given by

$$\min_{c_1, c_2, \Gamma} F^{CV}\,(c_1, c_2, \Gamma)\,. \tag{3.13}$$

In order to minimise (3.13), the author applied the level set method, where the unknown curve Γ is represented by the zero level set of the Lipschitz function ϕ such that

$$\Gamma = \{(x,y) \in \Omega : \phi(x,y) = 0\},$$
$$\text{inside}(\Gamma) = \{(x,y) \in \Omega : \phi(x,y) > 0\},$$
$$\text{outside}(\Gamma) = \{(x,y) \in \Omega : \phi(x,y) < 0\}.$$

Thus, the unknown lower dimensional variable curve Γ is replaced by another unknown higher dimensional variable ϕ. To reformulate (3.12), they defined the Heaviside function H and the Dirac delta function δ concentrated at 0, respectively by

$$H(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \qquad \text{and} \qquad \delta(x) = H'(x).$$

Expressing each term of the energy F in terms of ϕ give

$$\begin{aligned}
\text{Length}(\Gamma) &= \int_\Omega |\nabla H(\phi)| \, dxdy = \int_\Omega \delta(\phi) |\nabla \phi| \, dxdy, \\
\text{Area}(\text{inside}(\Gamma)) &= \int_\Omega H(\phi) \, dxdy \\
\int_{\text{inside}(\Gamma)} (z-c_1)^2 dxdy &= \int_\Omega (z-c_1)^2 H(\phi) \, dxdy \\
\int_{\text{outside}(\Gamma)} (z-c_2)^2 dxdy &= \int_\Omega (z-c_1)^2 (1 - H(\phi)) \, dxdy.
\end{aligned}$$

In the level set formulation, equation (3.12) is rewritten in the following way

$$\begin{aligned}
&F^{CV}(\phi, c_1, c_2) = \mu \int_\Omega \delta(\phi) |\nabla \phi| \, dxdy + v \int_\Omega H(\phi) \, dxdy \\
&+\lambda_1 \int_\Omega (z-c_1)^2 H(\phi) \, dxdy + \lambda_2 \int_\Omega (z-c_2)^2 (1 - H(\phi)) \, dxdy.
\end{aligned} \tag{3.14}$$

In implementation of CV model, both H and δ functions shall be regularised because H is not differentiable at 0. Both regularised functions are given as

$$H_\varepsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi} \arctan\left(\frac{x}{\varepsilon}\right)\right) \qquad \text{and} \qquad \delta_\varepsilon(x) = H'_\varepsilon(x) = \frac{\varepsilon}{\pi(\varepsilon^2 + x^2)} \tag{3.15}$$

Then the regularised functional denoted F_ε^{CV} by will be

$$\begin{aligned}
&F_\varepsilon^{CV}(\phi, c_1, c_2) = \mu \int_\Omega \delta_\varepsilon(\phi) |\nabla \phi| \, dxdy + v \int_\Omega H_\varepsilon(\phi) \, dxdy \\
&+\lambda_1 \int_\Omega (z-c_1)^2 H_\varepsilon(\phi) \, dxdy + \lambda_2 \int_\Omega (z-c_2)^2 (1 - H_\varepsilon(\phi)) \, dxdy.
\end{aligned} \tag{3.16}$$

with the new minimisation problem is given by

$$\min_{c_1, c_2, \Gamma} F_\varepsilon^{CV}(c_1, c_2, \Gamma). \tag{3.17}$$

Keeping the level set function φ fixed and minimizing equation (3.16) with respect to

c_1 and c_2, we have

$$c_1(\phi) = \frac{\int_\Omega z\,(x,y)\,H_\varepsilon\,(\phi)\;\,dxdy}{\int_\Omega H_\varepsilon\,(\phi)\;\,dxdy}, \qquad c_2(\phi) = \frac{\int_\Omega z\,(x,y)\,(1-H_\varepsilon\,(\phi))\;\,dxdy}{\int_\Omega (1-H_\varepsilon\,(\phi))\;\,dxdy} \tag{3.18}$$

After that, by fixing constants c_1 and c_2 and by first variations with respect to ϕ the authors derive the following Euler Lagrange equation for ϕ:

$$\begin{cases} \mu\delta_\varepsilon\,(\phi)\,\nabla\cdot\left(\dfrac{\nabla\phi}{|\nabla\phi|}\right) - v - \lambda_1\delta_\varepsilon\,(\phi)\,(z-c_1)^2 + \lambda_2\delta\,(\phi)\,(z-c_2)^2 = 0, \quad in \quad \Omega \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \dfrac{\partial\phi}{\partial\vec{n}} = 0, \quad on \quad \Omega \end{cases} \tag{3.19}$$

Details of the derivation of the Euler Lagrange equation and its solution using gradient descent scheme are given in the original paper [38]. To prevent the level set function ϕ becomes too steep or too flat, the re-initialise step of the level set function as a signed distance function to its zero level set as discussed in Section 2.7 can be done by solving the following PDE for ϕ

$$\begin{aligned} &\phi_t - \mathrm{sgn}\,(\phi)\,(1-|\nabla\phi|) = 0 \\ &\phi\,(\mathbf{x},0) = \phi_0 \end{aligned} \tag{3.20}$$

The PDE is proposed by Sussman et. al [131] where ϕ_0 is the function which is supposed to be re-initialised and $\mathrm{sgn}(\cdot)$ is a signed function. The solution of this equation, ϕ will have the same zero level set as ϕ_0 and away from this level set, $|\nabla\phi|$ will converge to 1, as it should be for a distance function. The numerical approximation for equation (3.20) is given by

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n - \Delta\tau sign\,(\phi\,(x,y,t))\,G\left(\phi_{i,j}^n\right)$$

where $G(\phi_{i,j}^n)$ is defined by

$$G\left(\phi_{i,j}^n\right) = \begin{cases} \sqrt{\max\left((a^+)^2,(b^-)^2\right) + \max\left((c^+)^2,(d^-)^2\right) - 1}, & \phi\,(x_i,y_j,t) > 0 \\ \sqrt{\max\left((a^-)^2,(b^+)^2\right) + \max\left((c^-)^2,(d^+)^2\right) - 1}, & \phi\,(x_i,y_j,t) < 0 \\ 0, & \text{otherwise} \end{cases}$$

Here,

$$
\begin{aligned}
a &= \frac{\left(\Delta_{-}^{x}\phi_{i,j}\right)}{h} = \frac{\left(\phi_{i,j}-\phi_{i-1,j}\right)}{h} \\
b &= \frac{\left(\Delta_{+}^{x}\phi_{i,j}\right)}{h} = \frac{\left(\phi_{i+1,j}-\phi_{i,j}\right)}{h} \\
c &= \frac{\left(\Delta_{-}^{y}\phi_{i,j}\right)}{h} = \frac{\left(\phi_{i,j}-\phi_{i,j-1}\right)}{h} \\
d &= \frac{\left(\Delta_{+}^{y}\phi_{i,j}\right)}{h} = \frac{\left(\phi_{i,j+1}-\phi_{i,j}\right)}{h}
\end{aligned}
$$

and $a^{+}=\max(a,0)$, $a^{-}=\min(a,0)$, and so on.

We remark that because the energy functional of CV is nonconvex (allowing therefore many local minima) the solution may depend on the initial curve in some cases. To avoid this drawback new variational models and techniques have been proposed, which we will overview in the following section.

3.3.5 Global Minimisation of the Active Contour Model

To overcome the presence of local minima and dependence of initialisation issue of CV model [38] due to nonconvexity of the functional, Chan et al. [34] proposed an algorithm to find the global minimisation of the model. First, we recall the Chan-Vese model

$$
\begin{aligned}
&\min_{\phi,c_1,c_2} F_{\varepsilon}^{CV}\left(\phi,c_1,c_2\right)=\mu\int_{\Omega}\delta_{\varepsilon}\left(\phi\right)\left|\nabla\phi\right|dxdy+v\int_{\Omega}H_{\varepsilon}\left(\phi\right)\,dxdy \\
&+\lambda_1\int_{\Omega}\left(z-c_1\right)^2H_{\varepsilon}\left(\phi\right)\,dxdy+\lambda_2\int_{\Omega}\left(z-c_2\right)^2\left(1-H_{\varepsilon}\left(\phi\right)\right)\,dxdy.
\end{aligned}
\tag{3.21}
$$

This minimisation problem is nonconvex due to the length term $\int_{\Omega}\left|\nabla H(\phi)\right|dxdy$ where the minimisation is carried out over a nonconvex set of function [34]. The approximation of the solution is obtained by a two step scheme where c_1 and c_2 are firstly computed and the second step is to update the curve with the gradient descent equation

$$
\frac{\partial\phi}{\partial t}=\delta_{\varepsilon}\left(\phi\right)\left[\nabla\cdot\left(\frac{\nabla\phi}{\left|\nabla\phi\right|}\right)-\lambda r\left(x,y\right)\right], \tag{3.22}
$$

where $r(x,y)=(z-c_1)^2-(z-c_2)^2$. In [38], the CV algorithm used a noncompactly supported, smooth approximation H_{ε} of Heaviside function H. Thus the above gradient descent equation (3.22) has the same stationary solution for

$$
\frac{\partial\phi}{\partial t}=\nabla\cdot\left(\frac{\nabla\phi}{\left|\nabla\phi\right|}\right)-\lambda r\left(x,y\right). \tag{3.23}
$$

Consequently, the equation (3.23) is the gradient descent equation of the following convex energy functional

$$
\int_{\Omega}\left|\nabla\phi\right|\,dxdy+\lambda\int_{\Omega}r\left(x,y\right)\phi\,dxdy. \tag{3.24}
$$

The functional (3.24) in general does not have a minimiser because it is homogeneous of degree 1 in ϕ, unless the minimisation to ϕ is restricted such that $0 \leq \phi \leq 1$, $\forall (x, y) \in \Omega$; see [34] for more details. Thus, the following minimisation problem is considered

$$\begin{aligned} &\min_{0 \leq \phi \leq 1} F^{GCV}(\phi, c_1, c_2), \\ &F^{GCV}(\phi, c_1, c_2) = \int_{\Omega} |\nabla \phi| \, dxdy + \lambda \int_{\Omega} r(x, y) \, \phi \, dxdy. \end{aligned} \tag{3.25}$$

With the restriction of $0 \leq \phi \leq 1$, $\forall (x, y) \in \Omega$ and following the work of Strang [130], the minimisation (3.25) leads to the global minimiser from the following theorem:

Theorem 3.3.1. *For any given fixed $c_1, c_2 \in \mathbb{R}$, a global minimiser for $F^{GCV}(u, c_1, c_2)$ can be found by carrying out the following convex minimisation*

$$\min_{0 \leq u \leq 1} \left\{ \int_{\Omega} |\nabla u| \, dxdy + \lambda \int_{\Omega} r(x, y) \, u \, dxdy \right\}, \tag{3.26}$$

and then setting $\sum = \{x : u(x, y) \geq \mu\}$, $\mu \in [0, 1]$.

In contrast to CV model, the theorem shows that the minimisation of (3.26) removes the non-convex constraint of being binary and instead the minimisation is carried out over functions that can take intermediate values. Chan et al. [34] further changed the minimization (3.26) into an unconstrained minimisation problem according to the following theorem:

Theorem 3.3.2. *Let $r(x, y) \in L^{\infty}(\Omega)$, for any given fixed $c_1, c_2 \in \mathbb{R}$ and $\lambda > 0$. Then the convex constrained minimisation* (3.26) *has the set of minimisers as the following convex unconstrained minimisation problem, a global minimiser for $F^{GCV}(u, c_1, c_2)$ can be found by carrying out the following convex minimisation*

$$\min_{u} \left\{ \int_{\Omega} |\nabla u| \, dxdy + \lambda \int_{\Omega} r(x, y) \, u \, dxdy + \alpha \int_{\Omega} \nu(u) \, dxdy \right\}, \tag{3.27}$$

where $\nu(\zeta) := \max\{0, 2|\zeta - 1/2| - 1\}$ is an exact penalty function, provided that the constant $\alpha > \frac{\lambda}{2} \|r(x, y)\|_{L^{\infty}(\Omega)}$.

The proof of the theorems and further details can be found in the paper [34]. The associate Euler Lagrange equation is given by

$$\frac{\partial \phi}{\partial t} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda r(x, y) - \alpha \nu'(\phi). \tag{3.28}$$

3.3.6 Geodesic Aided CV Model

The Geodesic active contour [25] in Section 3.33 is formulated based on gradient to detect object boundary in which only local information of boundary is used. For images with fuzzy, discrete edges, and very noisy, it is difficult to get desirable result. As the CV model [38] depends on the image information derived from homogenous regions, it can overcome the limitation of Geodesic active contour. However, despite these advantages, CV model has an unavoidable limitation. First, the formulation is restricted

to only two phases; the object and the background, resulting in problems in detecting more than two objects or multiple objects with complex background. Moreover, the precise boundary usually cannot be obtained as the method is based on information of homogeneous regions instead of local information. Due to these behaviour of both models, L. Chen et al. [41] introduced a new method called Geodesic Aided CV Model to address the above problems by combining Geodesic active contour and CV model in the following formulation

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \left[\mu \nabla \cdot \left(g \frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (z - c_1)^2 + \lambda_2 (z - c_2)^2 \right].$$

Since we have

$$\nabla \cdot \left(g \frac{\nabla \phi}{|\nabla \phi|} \right) = g \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \nabla g \cdot \frac{\nabla \phi}{|\nabla \phi|},$$

after some manipulations and approximations, they solve the following PDE

$$\frac{\partial \phi}{\partial t} = g\,|\nabla \phi| \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (z - c_1)^2 + \lambda_2 (z - c_2)^2 \right] + \nabla g \cdot \nabla z.$$

The first term on the right is called the region detector that combined the global and local information of an image in order to obtain accurate boundaries. The second term is called the local detector which attracts the evolving curve to the real boundary of object [41]. They also extended their method to colour images.

3.3.7 Geometrical Constraint Segmentation Model

It is common that the image data is missing or of poor quality or two objects are very close to each other. This result in difficulty to clearly identify the interface between them. The additional geometrical constraint added to the formulation can help the segmentation process. C. Gout et al. [65, 67] propose a new model for segmentation of an image under geometrical constraints to detect special features in an image. Due to this attribute, this model is considered as a selective segmentation model.

They have combined the Geodesic active contour model [25] with additional of geometrical constraints that is a set of points near the boundary of targeted object. Let $z(x, y)$ be a given image defined on Ω . Let $A = \{(x_i^*, y_i^*) \in \Omega,\ 1 \leq i \leq n_1\} \subset \Omega$ be the set of n_1 distinct points near the boundary of targeted object in $z(x, y)$. The objective is to determine an optimal contour $\Gamma \subset \Omega$ that best approaches the points from the set A and at the same time detecting the desired object in an image. The functions required to stop the evolving curve is the function g that is the edge detector function as in (3.7) and the function d defined as

$$\forall\,(x, y) \in \Omega, \qquad d\,(x, y) = \prod_{i=1}^{n_1} \left(1 - e^{-\frac{(x - x_i^*)^2}{2\sigma^2}} e^{-\frac{(y - y_i^*)^2}{2\sigma^2}} \right). \tag{3.29}$$

The function g is zero on edges in an image and is 1 in flat regions, while the function d acts locally and will be approximately 0 in the neighbourhood of points of

A. Then, a contour Γ is expected to stop at local minima which has $d \approx 0$ or $g \approx 0$ along it. The following functional is proposed

$$F^{GC}(\Gamma) = \int_{\Gamma} d(x,y)\, g(|\nabla z(x,y)|)\, ds. \tag{3.30}$$

The level set formulation is used in order to extend the domain of integral in (3.30) to the whole image other than Γ. Hence, Γ is considered as the zero level set of ϕ i.e

$$\Gamma = \{(x,y) \in \Omega : \phi(x,y) = 0\},$$

with $\phi < 0$ inside Γ and $\phi > 0$ outside Γ. Then, (3.30) is transformed into level sets formulation as follows

$$F^{GC}(\phi) = \int_{\Omega} d(x,y)\, g(|\nabla z(x,y)|)\, |\nabla H(\phi(x,y))|\, dxdy. \tag{3.31}$$

Here H is the Heaviside function and $\int_{\Omega} |\nabla H(\phi)|\, dxdy$ is the length of Γ. Thus we have the following minimisation problem

$$\min_{\phi(x,y)} F^{GC}(\phi(x,y)). \tag{3.32}$$

In practice, the regularised version of Heaviside function, H_ε is used because the original Heaviside function is nondifferentiable at 0. Thus the following minimisation problem is considered

$$\min_{\phi(x,y)} F_{\varepsilon}^{GC}(\phi(x,y)), \tag{3.33}$$

where

$$F_{\varepsilon}^{GC}(\phi) = \int_{\Omega} d(x,y)\, g(|\nabla z(x,y)|)\, |\nabla H_{\varepsilon}(\phi(x,y))|\, dxdy. \tag{3.34}$$

The following Euler Lagrange equation is derived and solved using AOS method

$$-\delta_{\varepsilon}(\phi(x,y))\, \nabla \cdot \left(d(x,y)\, g(|\nabla z(x,y)|)\, \frac{\nabla \phi(x,y)}{|\nabla \phi(x,y)|} \right) = 0. \tag{3.35}$$

3.4 Summary

In this chapter we mainly reviewed variational models for image segmentations. We first discussed the seminal work of ROF model [115] in image denoising mainly because the ROF model [115] provides the basis for the variational formulation of segmentation models of our particular interest in this study. The ROF model used the total variation (TV) regularisation in its formulation. Due to the effectiveness of TV regularisation, it is widely used in the formulation of image segmentation models where we have introduced seven image segmentation models in variational framework that related to our work; Mumford-Shah model [102], Snake model [77], Geodesic Active Contour [25] , Chan and

Vese model [38], global minimisation model [34], Geodesic Aided CV Model [41], and lastly the selective segmentation model namely Geometrical Constraint Segmentation Model [65, 67]. All the segmentation models are initially solved in optimise-discretise scheme where the associate PDE (Euler Lagrange equations) derived from the objective functionals are solved. This motivates us to develop an alternative numerical technique in discretise-optimise scheme in the next chapter where the objective functional is directly solved.

Chapter 4

An Optimisation based Multilevel Algorithm for Variational Image Segmentation Models

In this chapter we propose an optimisation based multilevel algorithm for efficiently solving a class of selective segmentation models. In level set function formulation, our first variant of the proposed multilevel algorithm has the expected optimal $O(N \log N)$ efficiency for an image of size $n \times n$ with $N = n^2$. However, modified localised models are proposed to exploit the local nature of segmentation contours and consequently our second variant after modification is up to practically super-optimal efficiency of $O(\sqrt{N} \log N)$. Numerical results show that good segmentation quality is obtained and excellent efficiency is observed in reducing the computational time.

4.1 Introduction

In the previous chapters we mentioned different techniques developed for image segmentation such as statistical methods [45, 44, 58], wavelet techniques [90], histogram analysis and thresholding [88, 121], variational edge detection active contours [77, 25, 123, 7] and region-based active contours [143, 102, 18, 1, 69, 16, 126, 38]

Segmentation models described above are for global segmentation due to the fact that all features or objects in an image are to be segmented. This chapter is concerned with another type of image segmentation models, namely selective segmentation. They are defined as the process of extracting one object of interest in an image based on some known geometric constraints [65, 111, 129]. Two effective models are Badshah-Chen [12] and Rada-Chen [111] which used a mixture of edge-based and region-based ideas in addition to imposing constraints. The additive operator splitting (AOS) method (suitable for images of moderate size, faster than gradient type methods) was proposed for such models. However, to process images of large size, urgent need exists in

developing fast multilevel methods.

Both the multilevel and multigrid methods are developed using the idea of hierarchy of discretisations. However, a multilevel method is based on discretise-optimise scheme (algebraic) where minimisation of a variational problem is solved directly without using a partial differential equation (PDE). In contrast, a multigrid method is based on optimise-discretise scheme (geometric) where it solves a PDE numerically. The two methods are inter-connected since both can have geometric interpretations and use similar inter-level information transfers.

The latter multigrid methods have been used to solve a few variational image segmentation models in the level set formulation. For geodesic active contours models, linear multigrid methods have been developed [78, 108, 109]. In 2008, Badshah and Chen [10] have successfully implemented a multigrid method to solve the Chan-Vese nonlinear elliptical partial differential equation. In 2009, Badshah and Chen [11] also have developed two multigrid algorithms for modelling variational multiphase image segmentation. While the practical performance of these methods is good, however, they are sensitive to parameters and hence not effective, mainly due to non-smooth coefficients which lead to smoothers not having an acceptable smoothing rate (which in turn are due to jumps or edges that separate segmented domains). Therefore the above multigrid methods behave like the cascadic multigrids [100] where only one multigrid cycle is needed.

Here we pursue the former type of optimisation based multilevel methods, based on a discretise-optimise scheme where the minimisation is solved directly (without using PDEs). The idea has been applied to other image problems in denoising and debluring [33, 29, 30], not yet to selective segmentation problems. However, the method is found to get stuck to local minima due to non-differentiability of the energy functional. To overcome that situation, Chan and Chen [33] have proposed the "patch detection" idea in the formulation of the multilevel method which is efficient for image denoising problems. However, as image size increases, the method can be slow because of the patch detection idea searches the entire image for the possible patch size on the finest level after each multilevel cycle.

In this work, we will consider a differentiable form of variational image segmentation models and develop the multilevel algorithm for the resulting models without using a "patch detection" idea. We are not aware of any similar work on multilevel algorithms for selective segmentation models in the level set formulation. The key finding is that the resulting multilevel algorithm converges, while not very sensitive to parameter choices, unlike geometric multigrid methods [11] which are known to have problems in convergence.

The rest of the chapter is organised in the following way. In Section 4.2, we briefly review two selective segmentation models which are Badshah-Chen model [12] and Rada-Chen model [111]. In Section 4.3, we present an optimisation based multilevel algorithm for the selective segmentation models. In Section 4.4, we propose localised segmentation models and further present multilevel methods for solving them in Section

4.5. In Section 4.6, we give some experimental results to test the presented algorithms. We compare the new methods to the previously fast methods from the literature namely the AOS method for Badshah-Chen [12] and Rada-Chen [111] models (since multigrid methods are not yet developed for these models). However, a multiscale AOS method (for Badshah-Chen [12] and Rada-Chen [111] models) based on the pyramid idea is implemented and included in the comparison.

The main idea of this chapter has been published by me as the main author together with my supervisor Prof. Ke Chen as the co-author in [73]. The multilevel algorithm and the multiscale AOS method for the selective segmentation models were developed and implemented in MATLAB by the author. The localised segmentation models were derived by the author. The multilevel algorithm for the localised models was jointly developed by us while the algorithm was implemented in MATLAB by the author.

4.2 Review of three existing models

In this section we will first briefly introduce the global segmentation model [38] because it provides the foundation for the selective segmentation models as well as a method for minimising the associated functional, details can be found in Chapter 3. Next, we will discuss two selective segmentation models by Badshah-Chen [12] and Rada-Chen [111] before we address the efficiency issue for these models.

4.2.1 The Chan-Vese model

The Chan and Vese (CV) model [38] considers a special case of the piecewise constant Mumford-Shah functional [102] where it is restricted to only two phases (i.e. constants), representing the foreground and the background of the given image $z(x, y)$.

Assume that z is formed by two regions of approximately piecewise constant intensities of distinct (unknown) values c_1 and c_2 , separated by some (unknown) curve or contour Γ. Let the object to be detected be represented by the region Ω_1 with the value c_1 inside the curve Γ whereas outside Γ, in $\Omega_2 = \Omega \backslash \Omega_1$, the intensity of z is approximated with the value c_2. Then, with $\Omega = \Omega_1 \cup \Omega_2$, the Chan-Vese model minimises the following functional

$$\begin{aligned} \min_{\Gamma, c_1, c_2} F_{CV}(\Gamma, c_1, c_2) = \mu \text{ length}(\Gamma) + \lambda_1 \int_{\Omega_1} (z - c_1)^2 dxdy \\ + \lambda_2 \int_{\Omega_2} (z - c_2)^2 dxdy. \end{aligned} \tag{4.1}$$

Here, the constants c_1 and c_2 are viewed as the average values of z inside and outside the variable contour Γ. The fixed parameters μ, λ_1, and λ_2 are non-negative but to be specified. In order to minimise equation (4.1), they applied the level set method [38], where the unknown curve Γ is represented by the zero level set of the Lipschitz

function such that

$$\begin{aligned}\Gamma &= \{(x,y) \in \Omega : \phi(x,y) = 0\},\\ \Omega_1 &= \text{inside}(\Gamma) = \{(x,y) \in \Omega : \phi(x,y) > 0\},\\ \Omega_2 &= \text{outside}(\Gamma) = \{(x,y) \in \Omega : \phi(x,y) < 0\}.\end{aligned}$$

To simplify the notation, denote the regularised versions of the Heaviside function and the Dirac delta function, respectively, by

$$H(\phi(x,y)) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\phi}{\varepsilon}\right)\right) \qquad \text{and} \qquad \delta(\phi(x,y)) = \frac{\varepsilon}{\pi(\varepsilon^2 + \phi^2)}.$$

Thus equation (4.1) becomes

$$\begin{aligned}&\min_{\phi,c_1,c_2} F_{CV}(\phi, c_1, c_2) = \mu\int_\Omega |\nabla H(\phi)|\, dxdy + \lambda_1\int_\Omega (z-c_1)^2 H(\phi)\, dxdy\\ &+\lambda_2\int_\Omega (z-c_2)^2 (1 - H(\phi))\, dxdy.\end{aligned} \tag{4.2}$$

Keeping the level set function ϕ fixed and minimising (4.2) with respect to c_1 and c_2, we have

$$c_1(\phi) = \frac{\int_\Omega z(x,y) H(\phi)\, dxdy}{\int_\Omega H(\phi)\, dxdy}, \qquad c_2(\phi) = \frac{\int_\Omega z(x,y)(1 - H(\phi))\, dxdy}{\int_\Omega (1 - H(\phi))\, dxdy}. \tag{4.3}$$

After that, by fixing constants c_1 and c_2 in $F_{CV}(\phi, c_1, c_2)$, first variation with respect to ϕ yields the following Euler-Lagrange equation:

$$\begin{cases} \mu\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda_1\delta(\phi)(z-c_1)^2 + \lambda_2\delta(\phi)(z-c_2)^2 = 0, \quad in \quad \Omega \\ \qquad\qquad \frac{\delta(\phi)}{|\nabla\phi|}\ \frac{\partial u}{\partial \vec{n}} = 0, \quad on \quad \partial\Omega. \end{cases} \tag{4.4}$$

Notice that the nonlinear coefficient in equation (4.4) may have a zero denominator, so the equation is not defined in such cases. A commonly-adopted idea to deal with $|\nabla\phi| = 0$ was to introduce a small positive parameter β to (4.2) and (4.4), so the new Euler Lagrange equation becomes

$$\begin{cases} \mu\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right) - \lambda_1\delta(\phi)(z-c_1)^2 + \lambda_2\delta(\phi)(z-c_2)^2 = 0, \quad in \quad \Omega \\ \qquad\qquad \frac{\delta(\phi)}{|\nabla\phi|}\ \frac{\partial u}{\partial \vec{n}} = 0, \quad on \quad \partial\Omega. \end{cases}$$

where corresponds to minimise the following differentiable energy function, instead of (4.2)

$$\begin{aligned}&\min_{\phi,c_1,c_2} F_{CV}(\phi, c_1, c_2) = \mu\int_\Omega \sqrt{|\nabla H(\phi)|^2 + \beta}\, dxdy\\ &+\lambda_1\int_\Omega (z-c_1)^2 H(\phi)\, dxdy + \lambda_2\int_\Omega (z-c_2)^2 (1 - H(\phi))\, dxdy.\end{aligned} \tag{4.5}$$

4.2.2 The Badshah-Chen model

The selective segmentation model by Badshah-Chen (BC) [12] combines the Geometric Constraint model of Gout et al [65, 67] (reviewed in Chapter 3) with intensity fitting terms of Chan-Vese [38]. For image $z(x,y)$ with a marker set $\mathcal{A} = \{w_i = (x_i^*, y_i^*) \in \Omega, 1 \leq i \leq n_1\} \subset \Omega$ of n_1 geometrical points on or near the target object [111, 152], the selective segmentation idea tries to detect the boundary of a single object among all homogeneity intensity objects in Ω closest to $\mathcal{A}$; here $n_1 \geq 3$. The geometrical points in $\mathcal{A}$ define an initial polygonal contour and guide its evolution towards Γ [152].

The BC minimisation equation [12] is given by

$$\begin{aligned} &\min_{\Gamma,c_1,c_2} F_{BC}(\Gamma, c_1, c_2) = \mu \int_{\Gamma} d(x,y)\, g(|\nabla z(x,y)|)\, dxdy \\ &+\lambda_1 \int_{\text{inside}(\Gamma)} (z-c_1)^2\, dxdy + \lambda_2 \int_{\text{outside}(\Gamma)} (z-c_2)^2\, dxdy. \end{aligned} \tag{4.6}$$

In this model, the function $g(|\nabla z|) = \frac{1}{1+\eta|\nabla z(x,y)|^2}$ is an edge detector which helps to stop the evolving curve on the edge of the targeted object. The strength of detection is adjusted by a parameter η. The function $g(|\nabla z|)$ is constructed to take small values near to 0 near object edges and large values near to 1 in flat regions. The $d(x,y)$ is a marker distance function which is close to 0 when approaching the points from marker set, given as:

$$d(x,y) = \text{distance}((x,y), \mathcal{A}) = \prod_{i=1}^{n_1} \left(1 - e^{-\frac{(x-x_i^*)^2}{2\kappa^2}} - e^{-\frac{(y-y_i^*)^2}{2\kappa^2}}\right), \quad \forall (x,y) \in \Omega$$

where κ is a positive constant. Alternative distance functions $d(x,y)$ are also possible [111, 152]. Using a level set formulation, the functional (4.6) becomes

$$\begin{aligned} &\min_{\phi,c_1,c_2} F_{BC}(\phi, c_1, c_2) = \mu \int_{\Omega} d(x,y) g(|\nabla z(x,y)|) |\nabla H(\phi)|\, dxdy \\ &+\lambda_1 \int_{\Omega} (z-c_1)^2 H(\phi)\, dxdy + \lambda_2 \int_{\Omega} (z-c_2)^2 (1-H(\phi))\, dxdy. \end{aligned} \tag{4.7}$$

Keeping the level set function ϕ fixed and minimising (4.6) with respect to c_1 and c_2, we have

$$c_1(\phi) = \frac{\int_\Omega z(x,y) H(\phi)\ dxdy}{\int_\Omega H(\phi)\ dxdy}, \qquad c_2(\phi) = \frac{\int_\Omega z(x,y)(1-H(\phi))\ dxdy}{\int_\Omega (1-H(\phi))\ dxdy}$$

Finally keeping constants c_1 and c_2 fixed in $F_{BC}(\phi, c_1, c_2)$, and the following Euler-Lagrange equation for ϕ is derived:

$$\begin{cases} \mu\delta(\phi)\nabla \cdot dg\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right) - \lambda_1\delta(\phi)(z-c_1)^2 & in \quad \Omega \\ +\lambda_2\delta(\phi)(z-c_2)^2 = 0, & \\ dg\frac{\delta(\phi)}{|\nabla\phi|}\ \frac{\partial u}{\partial \vec{n}} = 0, & on \quad \Omega \end{cases} \tag{4.8}$$

The small positive parameter β is introduced to avoid singularities in (4.8) which corresponds to minimise the following differentiable form of the BC model in replace of (4.7)

$$\begin{aligned}\min_{\phi,c_1,c_2} F_{BC}(\phi,c_1,c_2) &= \mu\int_\Omega G(x,y)\sqrt{|\nabla H(\phi)|^2+\beta}\,dxdy \\ &+\lambda_1\int_\Omega (z-c_1)^2 H(\phi)\,dxdy+\lambda_2\int_\Omega (z-c_2)^2(1-H(\phi))\,dxdy\end{aligned} \tag{4.9}$$

where $G = d(x,y)g(x,y)$.

4.2.3 The Rada-Chen model

The Rada-Chen (RC) model [111] imposes a further constraint on Ω_1 to ensure that its area is closest to the internal area defined by the marker set. From the polygon formed by the geometrical points in the set $\mathcal{A}$, denote by A_1 and A_2 respectively the area inside and outside the polygon. They compute A_1 and A_2 to approximate the area of the object they try to capture. The RC model also incorporates the similar edge detection function as in the BC model into the regularisation term. The energy minimisation problem is given by

$$\begin{aligned}\min_{\Gamma,c_1,c_2} F_{RC}(\Gamma,c_1,c_2) &= \mu\int_\Gamma g(|\nabla z(x,y)|)\,dxdy+\lambda_1\int_{\Omega_1}(z-c_1)^2dxdy \\ &+\lambda_2\int_{\Omega_2}(z-c_2)^2dxdy+\nu\left(\int_{\Omega_1}dxdy-\mathrm{A}_1\right)^2+\nu\left(\int_{\Omega_2}dxdy-\mathrm{A}_2\right)^2.\end{aligned} \tag{4.10}$$

Rewriting (4.10) in level-set formulation as in (4.2), we arrived at the following Euler-Lagrange equation for ϕ:

$$\begin{cases} \mu\delta(\phi)\nabla\cdot g\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right)-\lambda_1\delta(\phi)(z-c_1)^2 \\ \qquad +\lambda_2\delta(\phi)(z-c_2)^2-\nu\delta(\phi) & in \quad \Omega \\ \left[\left(\int_\Omega H(\phi)\,dxdy-\mathrm{A}_1\right)-\left(\int_\Omega(1-H(\phi))\,dxdy-\mathrm{A}_2\right)\right]=0, & \\ dg\frac{\delta(\phi)}{|\nabla\phi|}\,\frac{\partial u}{\partial\vec{n}}=0, & on \quad \Omega \end{cases} \tag{4.11}$$

As with the BC model, in the actual implementation of the RC model, the small positive parameter β is introduced to avoid singularities in (4.11) where corresponds to minimise the following differentiable form of the RC model

$$\begin{aligned}&\min_{\phi,c_1,c_2} F_{RC}(\phi,c_1,c_2)=\mu\int_\Omega g(|\nabla z(x,y)|)\sqrt{|\nabla H(\phi)|^2+\beta}\,dxdy+ \\ &\lambda_1\int_\Omega(z-c_1)^2H(\phi)\,dxdy+\lambda_2\int_\Omega(z-c_2)^2(1-H(\phi))\,dxdy+ \\ &\nu\left(\int_\Omega H(\phi)\,dxdy-\mathrm{A}_1\right)^2+\nu\left(\int_\Omega(1-H(\phi))\,dxdy-\mathrm{A}_2\right)^2.\end{aligned} \tag{4.12}$$

We will use the term **BC0** and **RCO** to refer the AOS algorithm previously used to solve BC model and RC model in [12] and [111] respectively.

Of course, it is known that such AOS method is not designed for processing large

image. To assist AOS, a pyramid method can be used. The basic idea in a pyramid method is, in the process of curve evolution, the pyramid scheme is used to decompose an image into different scale images and then coarse segmentation is performed on the coarse-scale image using the AOS method instead of directly using the original-size image. Then, the segmentation result is interpolated and adopted as an initial contour for the fine-scale image, thus gradually optimising the contour and reaching the final segmentation result. We refer the pyramid method for BC and RC models as **BCP** and **RCP** respectively.

The above forms of variational models, the BC model (4.9), the RC model (4.12) respectively, will be conveniently solved by our new proposed multilevel scheme shortly. The **BC0**, **RCO**, **BCP**, and **RCP** will be used as comparison methods to our method in segmenting large images.

As remarked before, the reason for seeking alternative optimisation based multilevel methods instead of applying a geometric multigrid method is that there are no effective smoothers for the latter case and consequently there exist no converging multigrid methods for the Euler-lagrange equations for our variational models.

4.3 An $O(N \log N)$ optimisation based multilevel algorithm

The main objective of this section is to present the first version of our multilevel formulation for two selective segmentation models: the BC model [12] and the RC model [111]. This section provides the foundation for the development of our main multilevel algorithm for the localised versions of these models. For simplicity for a given image of size $n \times n$, we shall assume $n = 2^L$. The standard coarsening defines $L+1$ levels: $k = 1$ (finest) $, 2, ..., L,\ L+1$ (coarsest) such that level k has $\tau_k \times \tau_k$ "superpixels" with each "superpixel" having pixels $b_k \times b_k$, where $\tau_k = n/2^{k-1}$ and $b_k = 2^{k-1}$. Figure 4.2(a-e) show the case of $L = 4,\ n = 2^4$ for an 16×16 image with 5 levels: level 1 has each pixel of the default size of 1×1 while the coarsest level 5 has a single superpixel of size 16×16. If $n \neq 2^L$, the multilevel method can still be developed with some coarse level superpixels of square shapes and the rest of rectangular shapes.

4.3.1 Multilevel algorithm for the BC model

Our goal is to solve (4.9), i.e. the BC model [12] , using a multilevel method in a discretise-optimise scheme.

Before we proceed further, one may question how to discretise the total variation (TV) term in the form

$$TV(u) = \int_{\Omega} |\nabla u| \, dxdy$$

TV is most often discretised by

$$TV_d(u) = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left(\nabla_x^+ u\right)_{i,j}^2 + \left(\nabla_y^+ u\right)_{i,j}^2}$$
$$= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left(u_{i+1,j}-u_{i,j}\right)^2 + \left(u_{i,j+1}-u_{i,j}\right)^2}$$

There are other ways to define discrete TV by finite difference, but the above form is the simplest one according to [97]. Furthermore, central differences are undesirable for TV discretisation (in a discretise-optimise approach) because they miss thin structure [59] as the central differences at (i,j) does not depend on $u_{i,j}$:

$$TV_d(u) = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left[\left(\nabla_x^+ u\right)/2 + \left(\nabla_x^- u\right)/2\right]^2 + \left[\left(\nabla_y^+ u\right)/2 + \left(\nabla_y^- u\right)/2\right]^2}$$
$$= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left[\left(u_{i+1,j}-u_{i-1,j}\right)/2\right]^2 + \left[\left(u_{i,j+1}-u_{i,j-1}\right)/2\right]^2}.$$

To avoid the problem, one sided difference can be used. More discussion on discretising TV can be found in [59] and [97] and the references therein.

Using the above information, the discretised version of (4.9) is given by:

$$\begin{aligned}\min_{\phi,c_1,c_2} F_{BC}(\phi,c_1,c_2) &\equiv \min_{\phi,c_1,c_2} F_{BC}^a(\phi_{1,1},\phi_{2,1},\ldots,\phi_{i-1,j},\phi_{i,j},\phi_{i+1,j},\ldots,\phi_{n,n},c_1,c_2)\\ &= \bar{\mu}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} G_{i,j}\sqrt{\left(H_{i,j}-H_{i,j+1}\right)^2 + \left(H_{i,j}-H_{i+1,j}\right)^2 + \beta}\\ &+ \quad \lambda_1\sum_{i=1}^{n}\sum_{j=1}^{n}(z_{i,j}-c_1)^2 H_{i,j} + \lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}(z_{i,j}-c_2)^2(1-H_{i,j})\end{aligned} \tag{4.13}$$

where ϕ denotes a row vector, $\bar{\mu} = \dfrac{\mu}{h}$, , $c_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} z_{i,j}H_{i,j} \Big/ \sum_{i=1}^{n}\sum_{j=1}^{n} H_{i,j}$, $G_{i,j} = G(x_i,y_j)$, $c_2 = \sum_{i,j=1}^{n} z_{i,j}(1-H_{i,j}) \Big/ \sum_{i,j=1}^{n}(1-H_{i,j})$ and $H_{i,j} = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\phi_{i,j}}{\varepsilon}\right)$.

As a prelude to multilevel methods, consider the minimisation of (4.13) by the coordinate descent method on the finest level 1:

$$\begin{cases} \text{Given} & \phi^{(0)} = \left(\phi_{i,j}^{(0)}\right) \text{ with } m = 0,\\ \text{Solve} & \phi_{i,j}^{(m+1)} = \underset{\phi_{i,j}\in\mathbb{R}}{\arg\min} F_{BC}^{loc}(\phi_{i,j},c_1,c_2) \text{ for } i,j = 1,2,...n,\\ & \text{Repeat the above steps with } m = m+1 \text{ until stopped.}\end{cases} \tag{4.14}$$

Here equation (4.14) is obtained by expanding and simplifying the main model in (4.13)

i.e.

$$
\begin{aligned}
&F^{loc}_{BC}(\phi_{i,j}, c_1, c_2) \\
&\equiv F^{a}_{BC}\left(\phi^{(m)}_{1,1}, \phi^{(m)}_{2,1}, \ldots, \phi^{(m)}_{i-1,j}, \phi_{i,j}, \phi^{(m)}_{i+1,j}, \ldots, \phi^{(m)}_{n,n}, c_1, c_2\right) - F^{(m)}_{BC} \\
&= \bar{\mu}\Big[G_{i,j}\sqrt{\left(H_{i,j} - H^{(m)}_{i+1,j}\right)^2 + \left(H_{i,j} - H^{(m)}_{i,j+1}\right)^2 + \beta} \\
&\quad + G_{i-1,j}\sqrt{\left(H_{i,j} - H^{(m)}_{i-1,j}\right)^2 + \left(H^{(m)}_{i-1,j} - H^{(m)}_{i-1,j+1}\right)^2 + \beta} \\
&\quad + G_{i,j-1}\sqrt{\left(H_{i,j} - H^{(m)}_{i,j-1}\right)^2 + \left(H^{(m)}_{i,j-1} - H^{(m)}_{i+1,j-1}\right)^2 + \beta}\Big] \\
&\quad + \lambda_1(z_{i,j} - c_1)^2 H_{i,j} \; + \lambda_2(z_{i,j} - c_2)^2\,(1 - H_{i,j})
\end{aligned}
$$

with Neumann's boundary condition, where $F^{(m)}_{BC}$ denotes the sum of all terms in F^{a}_{BC} that do not involve $\phi_{i,j}$. Minimisation of $c_1,\ c_2$ follows as before. Clearly one seems that this is a coordinate descent method. As such the method will exhibit a functional decay property $F^{a}_{BC}(\phi^{(m+1)}) \leq F^{a}_{BC}(\phi^{(m)})$ from one substep to the next. It should be remarked that the formulation in (4.14) is based on the work in [24, 33].

Using (4.14), we illustrate the interaction of $\phi_{i,j}$ with its neighboring pixel on the finest level 1 in Figure 4.1. We will use this basic structure to develop a multilevel method.

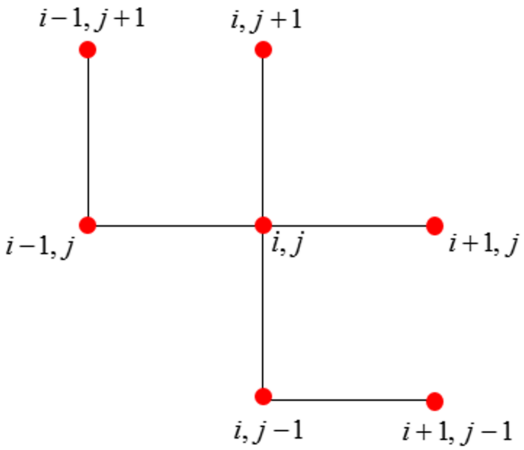


Figure 4.1: The interaction of $\phi_{i,j}$ at a central pixel (i, j) with neighboring pixels on the finest level 1. Clearly only 3 terms (pixels) are involved with $\phi_{i,j}$ (through regularisation).

The one-dimensional problem from (4.14) may be solved by any suitable optimisation method – here from $\phi^{(m)} \to \phi \to \phi^{(m+1)}$, we solve its first order condition

$$
\begin{aligned}
&\frac{\bar{\mu}G_{i,j}\left(2H_{i,j} - H^{(m)}_{i+1,j} - H^{(m)}_{i,j+1}\right)H'_{i,j}}{\sqrt{\left(H_{i,j} - H^{(m)}_{i+1,j}\right)^2 + \left(H_{i,j} - H^{(m)}_{i,j+1}\right)^2 + \beta}} + \frac{\bar{\mu}G_{i-1,j}\left(H_{i,j} - H^{(m)}_{i-1,j}\right)H'_{i,j}}{\sqrt{\left(H_{i,j} - H^{(m)}_{i-1,j}\right)^2 + \left(H^{(m)}_{i-1,j} - H^{(m)}_{i-1,j+1}\right)^2 + \beta}} \\
&+ \frac{\bar{\mu}G_{i,j-1}\left(H_{i,j} - H^{(m)}_{i,j-1}\right)H'_{i,j}}{\sqrt{\left(H_{i,j} - H^{(m)}_{i,j-1}\right)^2 + \left(H^{(m)}_{i,j-1} - H^{(m)}_{i+1,j-1}\right)^2 + \beta}} + H'_{i,j}\left(\lambda_1(z_{i,j} - c_1)^2 - \lambda_2(z_{i,j} - c_2)^2\right) = 0
\end{aligned}
$$

As an example, if using the Newton iterations, one gets the form

$$\phi_{i,j}^{new} = \phi_{i,j}^{old} - T^{old}/B^{old} \tag{4.15}$$

where

$$\begin{aligned} T^{old} &= \frac{\bar{\mu}G_{i,j}\left(2H_{i,j}^{old}-H_{i+1,j}^{(m)}-H_{i,j+1}^{(m)}\right){H'}^{old}{}_{i,j}}{D} + \frac{\bar{\mu}G_{i-1,j}\left(H_{i,j}^{old}-H_{i-1,j}^{(m)}\right){H'}^{old}{}_{i,j}}{E} \\ &+ \frac{\bar{\mu}G_{i,j-1}\left(H_{i,j}^{old}-H_{i,j-1}^{(m)}\right){H'}^{old}{}_{i,j}}{F} + {H'}^{old}{}_{i,j}\ \left(\lambda_1(z_{i,j}-c_1)^2 - \lambda_2(z_{i,j}-c_2)^2\right) \end{aligned}$$

$$\begin{aligned} B^{old} &= \frac{\bar{\mu}G_{i-1,j}\left(H_{i,j}^{old}-H_{i-1,j}^{(m)}\right){H''}^{old}{}_{i,j}+\left({H'}^{old}{}_{i,j}\right)^2}{E} - \frac{\bar{\mu}G_{i-1,j}\left(H_{i,j}^{old}-H_{i-1,j}^{(m)}\right)^2\left({H'}^{old}{}_{i,j}\right)^2}{E^3} \\ &+ \frac{\bar{\mu}G_{i,j}\left(2H_{i,j}^{old}-H_{i+1,j}^{(m)}-H_{i,j+1}^{(m)}\right){H''}^{old}{}_{i,j}+2\left({H'}^{old}{}_{i,j}\right)^2}{D} - \frac{\bar{\mu}G_{i,j}\left(2H_{i,j}^{old}-H_{i+1,j}^{(m)}-H_{i,j+1}^{(m)}\right)^2\left({H'}^{old}{}_{i,j}\right)^2}{D^3} \\ &+ \frac{\bar{\mu}G_{i,j-1}\left(H_{i,j}^{old}-H_{i,j-1}^{(m)}\right){H''}^{old}{}_{i,j}+\left({H'}^{old}{}_{i,j}\right)^2}{F} - \frac{\bar{\mu}G_{i,j-1}\left(H_{i,j}^{old}-H_{i,j-1}^{(m)}\right)^2\left({H'}^{old}{}_{i,j}\right)^2}{F^3} \\ &+ \left(\lambda_1(z_{i,j}-c_1)^2 - \lambda_2(z_{i,j}-c_2)^2\right){H''}^{old}{}_{i,j} \end{aligned}$$

and

$$\begin{aligned} D &= \sqrt{\left(H_{i,j}^{old}-H_{i+1,j}^{(m)}\right)^2 + \left(H_{i,j}^{old}-H_{i,j+1}^{(m)}\right)^2 + \beta} \\ E &= \sqrt{\left(H_{i,j}^{old}-H_{i-1,j}^{(m)}\right)^2 + \left(H_{i-1,j}^{(m)}-H_{i-1,j+1}^{(m)}\right)^2 + \beta} \\ F &= \sqrt{\left(H_{i,j}^{old}-H_{i,j-1}^{(m)}\right)^2 + \left(H_{i,j-1}^{(m)}-H_{i+1,j-1}^{(m)}\right)^2 + \beta} \end{aligned}$$

To develop a multilevel method of this coordinate descent method, we may interpret solving (4.14) for a new iterate $\phi_{i,j}^{(m+1)}$ as looking for the best update (on an old iterate $\phi_{i,j}^{(m)}$; here a scalar constant) that minimises the local merit functional $F_{BC}^{loc}(\phi_{i,j}, c_1, c_2)$. On level 1 the local minimisation for c takes the form

$$F_{BC}^{loc}(\phi_{i,j}, c_1, c_2) = F_{BC}^{loc}\left(\phi_{i,j}^{(m)} + c, c_1, c_2\right).$$

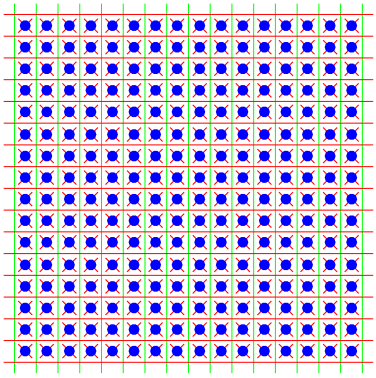
Hence, we may rewrite (4.14) in an equivalent form:

$$\begin{cases} \text{Given} & \phi^{(0)} = \left(\phi_{i,j}^{(0)}\right) \ \text{ with } m = 0, \\ \text{Solve} & \hat{c} = \arg\min_{c\in\mathbb{R}} F_{BC}^{loc}\left(\phi_{i,j}^{(m)} + c, c_1, c_2\right) \ \text{ for } i,j = 1,2,...n, \\ \text{Update} & \phi_{i,j}^{(m+1)} = \phi_{i,j}^{(m)} + \hat{c}, \\ & \text{Repeat the above steps with } m = m+1 \text{ until stopped.} \end{cases} \tag{4.16}$$

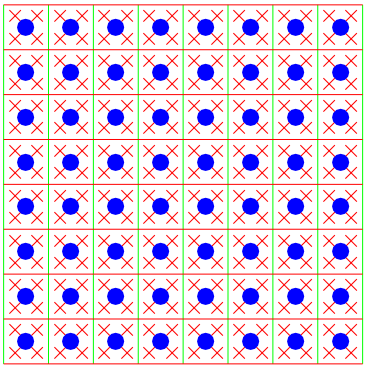
Now consider how the update is done on a general level $k = 2, \ldots, L+1$. Similarly to $k = 1$, we derive the simplified formulation for each of the $\tau_k \times \tau_k$ subproblems, in a block of pixels $b_k \times b_k$ e.g. the multilevel method for k=2 is to look for the best correction constant to update this block so that the underlying merit functional, relating to all four pixels (see Figure 4.2(b)), achieves a local minimum.

For levels $k = 1, ..., 5$, Figure 4.2 illustrates the multilevel partition of an image of

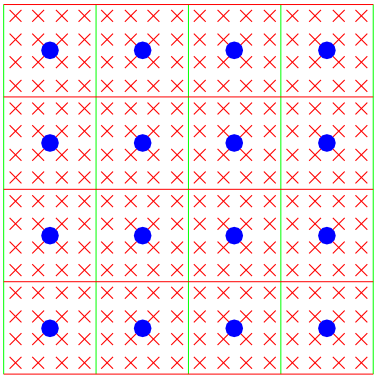
size 16×16 pixels from (a) the finest level (level 1) until (e) the coarsest level (level 5).



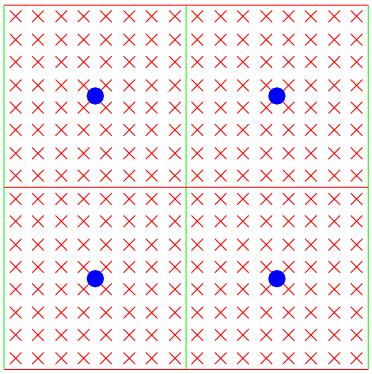
(a) Level 1: $\tau_1^2 = 16^2$ variables



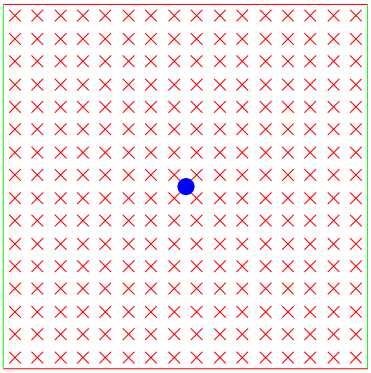
(b) Level 2: $\tau_2^2 = 8^2$ variables



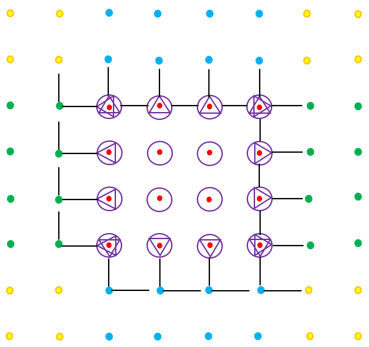
(c) Level 3: $\tau_3^2 = 4^2$ variables



(d) Level 4: $\tau_4^2 = 2^2$ variables



(e) Level 5: $\tau_5^2 = 1$ variable



(f) Level 3 block with $b_3^2 = 16$ pixels but only 12 effective terms in local minimization F_{BC}^{loc}

Figure 4.2: Illustration of multilevel coarsening. Partitions (a)-(e): the red "$\times$" shows image pixels, while blue $\bullet$ illustrates the variable c. (f) shows on coarse level 3 the difference of inner and boundary pixels interacting with neighboring pixels $\bullet$. The middle boxes $\odot$ indicate the inner pixels which do not involve c, others boundary pixels denoted by symbols $\lhd$, $\rhd$, Δ, ∇ involve c as in (4.16) via F_{BC}^{loc}.

Observe that $b_k \tau_k = n$ on level k, where τ_k is the number of boxes and b_k is the block size. So from Figure 1(a), $b_1 = 1$ and $\tau_1 = n = 16$. On other levels $k = 2, 3, 4$ and 5, we see that block size $b_k = 2^{k-1}$ and $\tau_k = 2^{L+1-k}$ since $n = 2^L$. Based on Figure 4.1, we illustrate a box $\odot$ interacting with neighboring pixels $\bullet$ in level 3. In addition, Figure

4.2 (f) illustrates that fact that variation by $c_{i,j}$ inside an active block only involves its boundary of precisely $4b_k - 4$ pixels, not all b_k^2 pixels, in that box, denoted by symbols $\lhd$, $\rhd$, Δ, ∇. This is important in efficient implementation.

With the above information, we are now ready to formulate the multilevel approach for general level k. Let's set the following: $b = 2^{k-1}$, $k_1 = (i-1)\,b + 1$, $k_2 = ib$, $\ell_1 = (j-1)\,b + 1$, $\ell_2 = jb$, and $c = (c_{i,j})$. Then, the computational stencil involving c on level k can be shown as follows

$$\begin{array}{c|c|c|c|c}
\vdots & \vdots & \cdots & \vdots & \vdots \\
\tilde{\phi}_{k_1-1,\ell_2+1} + c_{i-1,j+1} & \tilde{\phi}_{k_1,\ell_2+1} + c_{i,j+1} & \cdots & \tilde{\phi}_{k_2,\ell_2+1} + c_{i,j+1} & \tilde{\phi}_{k_2+1,\ell_2+1} + c_{i+1,j+1} \\
\hline
\tilde{\phi}_{k_1-1,\ell_2} + c_{i-1,j} & \tilde{\phi}_{k_1,\ell_2} + c_{i,j} & \cdots & \tilde{\phi}_{k_2,\ell_2} + c_{i,j} & \tilde{\phi}_{k_2+1,\ell_2} + c_{i+1,j} \\
\cdots & \vdots & \cdots & \vdots & \cdots \\
\tilde{\phi}_{k_1-1,\ell_1} + c_{i-1,j} & \tilde{\phi}_{k_1,\ell_1} + c_{i,j} & \cdots & \tilde{\phi}_{k_2,\ell_1} + c_{i,j} & \tilde{\phi}_{k_2+1,\ell_1} + c_{i+1,j} \\
\hline
\tilde{\phi}_{k_1-1,\ell_1-1} + c_{i-1,j-1} & \tilde{\phi}_{k_1,\ell_1-1} + c_{i,j-1} & \cdots & \tilde{\phi}_{k_2,\ell_1-1} + c_{i,j-1} & \tilde{\phi}_{k_2+1,\ell_1-1} + c_{i+1,j-1} \\
\vdots & \vdots & \cdots & \vdots & \vdots
\end{array} \tag{4.17}$$

The illustration shown above is consistent with Figure 4.2 (f) and the key point is that interior pixels (non-boundary pixels) do not involve $c_{i,j}$ in the formulation's first nonlinear term. This is because the finite differences are not changed at interior pixels by the same update as in

$$\begin{aligned}
&\sqrt{\left(\tilde{\phi}_{k,l} + c_{i,j} - \tilde{\phi}_{k+1,l} - c_{i,j}\right)^2 + \left(\tilde{\phi}_{k,l} + c_{i,j} - \tilde{\phi}_{k,l+1} - c_{i,j}\right)^2 + \beta} \\
&\qquad = \sqrt{\left(\tilde{\phi}_{k,l} - \tilde{\phi}_{k+1,l}\right)^2 + \left(\tilde{\phi}_{k,l} - \tilde{\phi}_{k,l+1}\right)^2 + \beta}.
\end{aligned}$$

Then, as a local minimisation for c, the problem (4.16) is equivalent to minimise the following

$$\begin{aligned}
F_{BC1}\left(c_{i,j}\right) =& \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell} \sqrt{\left[c_{i,j} - \left(\tilde{\phi}_{k_1-1,\ell} - \tilde{\phi}_{k_1,\ell}\right)\right]^2 + \left(\tilde{\phi}_{k_1-1,\ell} - \tilde{\phi}_{k_1-1,\ell+1}\right)^2 + \beta} \\
&+ \bar{\mu} \sum_{k=k_1}^{k_2-1} G_{k,\ell_2} \sqrt{\left[c_{i,j} - \left(\tilde{\phi}_{k,\ell_2+1} - \tilde{\phi}_{k,\ell_2}\right)\right]^2 + \left(\tilde{\phi}_{k,\ell_2} - \tilde{\phi}_{k+1,\ell_2}\right)^2 + \beta} \\
&+ \bar{\mu} G_{k_2,\ell_2} \sqrt{\left[c_{i,j} - \left(\tilde{\phi}_{k_2,\ell_2+1} - \tilde{\phi}_{k_2,\ell_2}\right)\right]^2 + \left[c_{i,j} - \left(\tilde{\phi}_{k_2+1,\ell_2} - \tilde{\phi}_{k_2,\ell_2}\right)\right]^2 + \beta} \\
&+ \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2,\ell} \sqrt{\left[c_{i,j} - \left(\tilde{\phi}_{k_2+1,\ell} - \tilde{\phi}_{k_2,\ell}\right)\right]^2 + \left(\tilde{\phi}_{k_2,\ell} - \tilde{\phi}_{k_2,\ell+1}\right)^2 + \beta} \\
&+ \bar{\mu} \sum_{k=k_1}^{k_2} G_{k,\ell_1-1} \sqrt{\left[c_{i,j} - \left(\tilde{\phi}_{k,\ell_1-1} - \tilde{\phi}_{k,\ell_1}\right)\right]^2 + \left(\tilde{\phi}_{k,\ell_1-1} - \tilde{\phi}_{k+1,\ell_1-1}\right)^2 + \beta} \\
&+ \lambda_2 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(1 - H\left(\tilde{\phi}_{k,\ell} + c_{i,j}\right)\right) \left(z_{k,\ell} - c_2\right)^2 \\
&+ \lambda_1 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(H\left(\tilde{\phi}_{k,\ell} + c_{i,j}\right)\right) \left(z_{k,\ell} - c_1\right)^2.
\end{aligned} \tag{4.18}$$

For the third term, we may note

$$\sqrt{(c-a)^2+(c-b)^2+\beta}=\sqrt{2\left(c-\frac{a+b}{2}\right)^2+2\left(\frac{a-b}{2}\right)^2+\beta}$$

.

Further we conclude that the local minimisation problem for block (i,j) on level k with respect to $c_{i,j}$ amounts to minimise the following equivalent functional

$$\begin{aligned}
F_{BC1}\left(c_{i,j}\right) =&\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell}\sqrt{(c_{i,j}-h_{k_1-1,\ell})^2+\upsilon_{k_1-1,\ell}^2+\beta}\\
&+\bar{\mu}\sum_{k=k_1}^{k_2-1} G_{k,\ell_2}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_2}\right)^2+h_{k,\ell_2}^2+\beta}\\
&+\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2,\ell}\sqrt{\left(c_{i,j}-h_{k_2,\ell}\right)^2+\upsilon_{k_2,\ell}^2+\beta}\\
&+\bar{\mu}\sum_{k=k_1}^{k_2} G_{k,\ell_1-1}\sqrt{(c_{i,j}-\upsilon_{k,\ell_1-1})^2+h_{k,\ell_1-1}^2+\beta}\\
&+\bar{\mu}\sqrt{2}G_{k_2,\ell_2}\sqrt{(c_{i,j}-\bar{\upsilon}_{k_2,\ell_2})^2+\bar{h}_{k_2,\ell_2}^2+\frac{\beta}{2}}\\
&+\lambda_1\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2} H(\tilde{\phi}_{k,\ell}+c_{i,j})(z_{k,\ell}-c_1)^2\\
&+\lambda_2\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)(z_{k,\ell}-c_2)^2
\end{aligned}\tag{4.19}$$

where we have used the following notation (which will be used later also):

$$\begin{array}{lll}
h_{k,\ell}=\tilde{\phi}_{k+1,\ell}-\tilde{\phi}_{k,\ell}, & \upsilon_{k,\ell}=\tilde{\phi}_{k,\ell+1}-\tilde{\phi}_{k,\ell}, & \upsilon_{k_2,\ell_2}=\tilde{\phi}_{k_2,\ell_2+1}-\tilde{\phi}_{k_2,\ell_2},\\
h_{k_2,\ell_2}=\tilde{\phi}_{k_2+1,\ell_2}-\tilde{\phi}_{k_2,\ell_2}, & \bar{\upsilon}_{k_2,\ell_2}=\frac{\upsilon_{k_2,\ell_2}+h_{k_2,\ell_2}}{2}, & \bar{h}_{k_2,\ell_2}=\frac{\upsilon_{k_2,\ell_2}-h_{k_2,\ell_2}}{2},\\
h_{k_1-1,\ell}=\tilde{\phi}_{k_1,\ell}-\tilde{\phi}_{k_1-1,\ell}, & \upsilon_{k_1-1,\ell}=\tilde{\phi}_{k_1-1,\ell+1}-\tilde{\phi}_{k_1-1,\ell}, & \upsilon_{k,\ell_2}=\tilde{\phi}_{k,\ell_2+1}-\tilde{\phi}_{k,\ell_2},\\
h_{k,\ell_2}=\tilde{\phi}_{k+1,\ell_2}-\tilde{\phi}_{k,\ell_2}, & h_{k_2,\ell}=\tilde{\phi}_{k_2+1,\ell}-\tilde{\phi}_{k_2,\ell}, & \upsilon_{k_2,\ell}=\tilde{\phi}_{k_2,\ell+1}-\tilde{\phi}_{k_2,\ell},\\
\upsilon_{k,\ell_1-1}=\tilde{\phi}_{k,\ell_1}-\tilde{\phi}_{k,\ell_1-1}, & h_{k,\ell_1-1}=\tilde{\phi}_{k+1,\ell_1-1}-\tilde{\phi}_{k,\ell_1-1}. &
\end{array}$$

On the coarsest level, we look for a **single** constant update for the current approximation $\tilde{\phi}$ that is $F_{BC1}\left(\tilde{\phi}+c\right)$,

$$\min_c F_{BC1}(\tilde{\phi}+c)=\bar{\mu}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} G_{i,j}\sqrt{\left(\tilde{\phi}_{i,j}+c-\tilde{\phi}_{i,j+1}-c\right)^2+\left(\tilde{\phi}_{i,j}+c-\tilde{\phi}_{i+1,j}-c\right)^2+\beta}$$

$$+\lambda_1\sum_{i=1}^{n}\sum_{j=1}^{n} H(\tilde{\phi}_{i,j}+c)(z_{i,j}-c_1)^2+\lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}\left(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)(z_{i,j}-c_1)^2$$

which is equivalent to

$$\begin{aligned}\min_c \hat{F}_{BC1}(\tilde{\phi}+c) = \lambda_1 \sum_{i=1}^{n}\sum_{j=1}^{n} H(\tilde{\phi}_{i,j}+c)(z_{i,j}-c_1)^2 \\ +\lambda_2 \sum_{i=1}^{n}\sum_{j=1}^{n} \big(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\big)(z_{i,j}-c_1)^2.\end{aligned} \tag{4.20}$$

The solutions of the above local minimisation problems, solved using a Newton method as in (4.15) or a fixed point method for t iterations (inner iteration), defines the updated solution $\tilde{\phi} = \tilde{\phi} + Q_k c$. Here Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_k \times b_k$ block on level k as illustrated in (4.17). Then we obtain a multilevel method if we cycle through all levels and all blocks on each level until the solution converges to the prescribe tolerance, *tol* or reach the prescribe maximum cycle (outer iteration).

So finally our implementation of the proposed multilevel method is then summarised in Algorithm 1. Here Steps $2-3$ simply update $\tilde{\phi}$ from the finest to the coarsest level

Algorithm 1 BC1 – Multilevel algorithm for the BC model

Given z, an initial guess $\tilde{\phi}$ and the stop tolerance tol with $L+1$ levels,

1) *Iteration starts with $\phi_{old} = \tilde{\phi}$ ($\tilde{\phi}$ contains the initial guess before the first iteration and the updated solution at all later iterations)*

2) *Smooth for t iterations the approximation on the finest level $k = 1$ that is solve* $\min_{\phi_{i,j}} F^{loc}_{BC}(\phi_{i,j}, c_1, c_2)$ or (4.14) *for $i, j = 1, 2, ...n$*

3) *Iterate for t times on each coarse level $k = 2, 3, ...L, L+1$:*
   - ▶ *If $k \leq L$, compute the minimiser c of* (4.19) or solve $\min_{c_{i,j}} F_{BC1}(c_{i,j})$;
   - ▶ *If $k = L+1$, solve* (4.20) or $\min_c \hat{F}_{BC1}(\tilde{\phi}+c)$ *on the coarsest level.*

   Add the correction $\tilde{\phi} = \tilde{\phi} + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_k \times b_k$ block on level k as illustrated in (4.17).

4) *Return to Step 1 unless $\frac{\|\tilde{\phi}-\phi_{old}\|_2}{\|\tilde{\phi}\|_2} < tol$ or until the prescribed maximum of cycles is reached. Otherwise exit with $\phi = \tilde{\phi}$.*

$k = 1, 2, \ldots, L, L+1$ so they might be viewed as a single step. We will use the term **BC1** to refer the multilevel Algorithm 1.

In this algorithm, we recommend that we start updating our multilevel algorithm in a fast manner is to adjust the fine structure before the coarse level. We found in a separate experiment that if we adjust the coarse structure before the fine level, the convergence is slower.

4.3.2 Multilevel algorithm for the RC model

Generalisation of the above algorithm to other models is much similar. For the RC model, the discretised version of in (4.12) takes the following form

$$\begin{aligned}
&\min_{\phi,c_1,c_2} F_{RC}\left(\phi,c_1,c_2\right)\\
&=\bar{\mu}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} g_{i,j}\sqrt{\left(H_{i,j}-H_{i,j+1}\right)^2+\left(H_{i,j}-H_{i+1,j}\right)^2+\beta}\\
&\lambda_1\sum_{i=1}^{n}\sum_{j=1}^{n}\left(z_{i,j}-c_1\right)^2H_{i,j}+\lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}\left(z_{i,j}-c_2\right)^2\left(1-H_{i,j}\right)\\
&+\nu\Big(-\mathrm{A}_1+\sum_{i=1}^{n}\sum_{j=1}^{n}H_{i,j}\Big)^2+\nu\Big(-\mathrm{A}_2+\sum_{i=1}^{n}\sum_{j=1}^{n}\Big(1-H_{i,j}\Big)\Big)^2.
\end{aligned}\tag{4.21}$$

Consider the minimisation of (4.21) by the coordinate descent method on the finest level 1:

$$\begin{cases}
\text{Given} & \phi^{(0)}=\left(\phi_{i,j}^{(0)}\right) \text{ with } m=0,\\
\text{Solve} & \phi_{i,j}^{(m+1)}=\underset{\phi_{i,j}\in\mathbb{R}}{\arg\min}\, F_{RC}^{loc}\left(\phi_{i,j},c_1,c_2\right) \text{ for } i,j=1,2,...n,\\
& \text{Repeat the above steps with } m=m+1 \text{ until stopped.}
\end{cases}\tag{4.22}$$

Here,

$$\begin{aligned}
F_{RC}^{loc}\left(\phi_{i,j},c_1,c_2\right)=&F_{RC}-F_0=\\
&\bar{\mu}\Big[g_{i,j}\sqrt{\left(H_{i,j}-H_{i+1,j}^{(m)}\right)^2+\left(H_{i,j}-H_{i,j+1}^{(m)}\right)^2+\beta}\\
&+g_{i-1,j}\sqrt{\left(H_{i,j}-H_{i-1,j}^{(m)}\right)^2+\left(H_{i-1,j}^{(m)}-H_{i-1,j+1}^{(m)}\right)^2+\beta}\\
&+g_{i,j-1}\sqrt{\left(H_{i,j}-H_{i,j-1}^{(m)}\right)^2+\left(H_{i,j-1}^{(m)}-H_{i+1,j-1}^{(m)}\right)^2+\beta}\Big]\\
&+\lambda_1(z_{i,j}-c_1)^2H_{i,j}+\lambda_2(z_{i,j}-c_2)^2(1-H_{i,j})\\
&+\nu(H_{i,j}-\mathrm{A}_1)^2+\nu((1-H_{i,j})-\mathrm{A}_2)^2.
\end{aligned}$$

The term F_0 refers to a collection of all terms that are not dependent on $\phi_{i,j}$. For $\phi_{i,j}$ at the boundary, Neumann's condition is used. In order to introduce the multilevel algorithm we first rewrite (4.22) in an equivalent form:

$$\hat{c}=\underset{c\in\mathbb{R}}{\arg\min}\, F_{RC}^{loc}\left(\phi_{i,j}^{(m)}+c,c_1,c_2\right),\ \phi_{i,j}^{(m+1)}=\phi_{i,j}^{(m)}+\hat{c} \text{ for } i,j=1,2,...,n.\tag{4.23}$$

Similar to BC1, we arrive at the following local functional for $\hat{c}$ on a general level:

$$\begin{aligned}
F_{RC1}\left(c_{i,j}\right) =& \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{\left(c_{i,j} - h_{k_1-1,\ell}\right)^2 + \upsilon_{k_1-1,\ell}^2 + \beta} \\
&+ \nu \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(-\mathrm{A}_1 + H(\tilde{\phi}_{k,\ell} + c_{i,j})\right)^2 \\
&+ \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{\left(c_{i,j} - h_{k_2,\ell}\right)^2 + \upsilon_{k_2,\ell}^2 + \beta} \\
&+ \bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_1-1}\right)^2 + h_{k,\ell_1-1}^2 + \beta} \\
&+ \bar{\mu}\sqrt{2} g_{k_2,\ell_2} \sqrt{\left(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2}\right)^2 + \bar{h}_{k_2,\ell_2}^2 + \frac{\beta}{2}} \\
&+ \lambda_1 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} H(\tilde{\phi}_{k,\ell} + c_{i,j}) (z_{k,\ell} - c_1)^2 \\
&+ \lambda_2 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} (1 - H(\tilde{\phi}_{k,\ell} + c_{i,j})) (z_{k,\ell} - c_2)^2 \\
&+ \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_2}\right)^2 + h_{k,\ell_2}^2 + \beta} \\
&+ \nu \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(-\mathrm{A}_2 + (1 - H(\tilde{\phi}_{k,\ell} + c_{i,j}))\right)^2. \qquad (4.24)
\end{aligned}$$

A single constant update on the current $\tilde{\phi}$ on the coarsest level is given by solving

$$\begin{aligned}
\min_c \hat{F}_{RC1}(\tilde{\phi} + c) =& \lambda_1 \sum_{i=1}^{n} \sum_{j=1}^{n} H(\tilde{\phi}_{i,j} + c)(z_{i,j} - c_1)^2 + \nu \sum_{i=1}^{n} \sum_{j=1}^{n} \Big(-\mathrm{A}_1 + H(\tilde{\phi}_{i,j} + c)\Big)^2 \\
&+ \lambda_2 \sum_{i=1}^{n} \sum_{j=1}^{n} \Big(1 - H(\tilde{\phi}_{i,j} + c)\Big) (z_{i,j} - c_2)^2 \\
&+ \nu \sum_{i=1}^{n} \sum_{j=1}^{n} \Big(-\mathrm{A}_2 + \Big(1 - H(\tilde{\phi}_{i,j} + c)\Big)\Big)^2. \qquad (4.25)
\end{aligned}$$

Our implementation of the proposed multilevel method is then summarised in Algorithm 2 which will be referred to as **RC1**.

Before we conclude this section, we give a brief convergence analysis of **BC1** and **RC1**. Let $N = n^2$ be the total number of pixels (unknowns). First, we compute the number of floating point operations (flops) for **BC1** for level k as follows:

Algorithm 2 RC1 – Multilevel algorithm for the RC model

Given z, an initial guess $\tilde{\phi}$ and the stop tolerance tol with $L+1$ levels,

1) *Iteration starts with $\phi_{old} = \tilde{\phi}$ ($\tilde{\phi}$ contains the initial guess before the first iteration and the updated solution at all later iterations)*

2) *Smooth for t iterations the approximation on the finest level 1 i.e. solve* $\min_{\phi_{i,j}} F^{loc}_{RC}(\phi_{i,j}, c_1, c_2)$ or (4.22) *for $i, j = 1, 2, ...n$*

3) *Iterate for t times on each coarse level $k = 2, 3, ...L, L+1$:*
   - ▶ *If $k \leq L$, compute the minimiser c of* (4.24) or solve $\min_{c_{i,j}} F_{RC1}(c_{i,j})$;
   - ▶ *If $k = L+1$, solve* (4.25) or $\min_c \hat{F}_{RC1}(\tilde{\phi} + c)$ *on the coarsest level .*

   Add the correction $\tilde{\phi} = \tilde{\phi} + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_k \times b_k$ block on level k as illustrated in (4.17).

4) *Return to Step 1 unless $\frac{\|\tilde{\phi} - \phi_{old}\|_2}{\|\tilde{\phi}\|_2} < tol$ or until the prescribed maximum of cycles is reached. Otherwise exit with $\phi = \tilde{\phi}$.*

Quantities	Flop counts for **BC1**
$h,\ \upsilon$	$4b_k\tau_k^2$
λ_1 term	$2N$
λ_2 term	$2N$
s smoothing steps	$38b_k\tau_k^2 s$

Then, the flop counts for all level is $\xi_{BC1} = \sum_{k=1}^{L+1} \left(4N + 4b_k\tau_k^2 + 38b_k\tau_k^2 s\right)$ where $k = 1$ (finest) and $k = L+1$ (coarsest). Next, we compute the upper bound for **BC1** as follows:

$$\xi_{BC1} = 4(L+1)N + \sum_{k=1}^{L+1}\left(\frac{4N}{b_k} + \frac{38Ns}{b_k}\right) = 4(L+1)N + (4+38s)N\sum_{k=0}^{L}\left(\frac{1}{2^k}\right)$$
$$< 4N\log n + 12N + 76Ns \approx O\left(N\log N\right)$$

Similarly, the flops for **RC1** is given as

Quantities	Flop counts for **RC1**
$h,\ \upsilon$	$4b_k\tau_k^2$
λ_1 term	$2N$
λ_2 term	$2N$
v term	$4N$
s smoothing steps	$31b_k\tau_k^2 s$

Hence, the total flop counts for **RC1** is $\xi_{RC1} = \sum_{k=1}^{L+1}\left(8N + 4b_k\tau_k^2 + 31b_k\tau_k^2 s\right)$. This gives the upper bound for **RC1** as

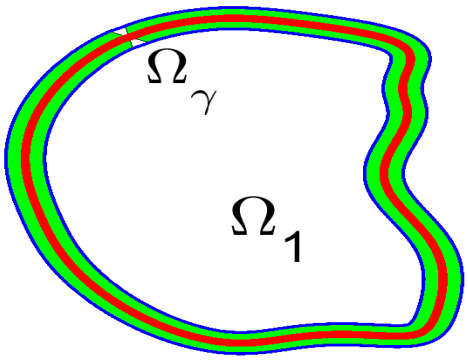


Figure 4.3: New modelling setup: replacement of domain Ω_1 by a smaller domain Ω_γ.

$$\xi_{RC1} = 8(L+1)N + \sum_{k=1}^{L+1}\left(\frac{4N}{b_k} + \frac{31Ns}{b_k}\right) = 8(L+1)N + (4+31s)N\sum_{k=0}^{L}\left(\frac{1}{2^k}\right)$$
$$< 8N\log n + 16N + 62Ns \approx O\left(N\log N\right)$$

One can observe that both **BC1** and **RC1** are of the optimal complexity $O(N\log N)$ expected of a multilevel method and $\xi_{RC1} > \xi_{BC1}$.

It may be remarked that both algorithms **BC1** and **RC1** are easily paralellisable and hence there is much potential to explore parallel efficiency. However below we consider how to improve the sequence efficiency in a simple and yet effective manner.

4.4 The new localised models

The complexity of the above presented algorithms is $O(N\log N)$ per cycle through all levels, for an image sized $n\times n$ with $N = n^2$. As this is optimal, for most problems, there is no need to consider further reduction for many problems e.g. for image denoising. However, segmentation is a special problem because evolution of level set functions ϕ is always local in selective segmentation. Below we turn this locality into reformulations and explore further reduction of the $O(N\log N)$ complexity, consequently achieving the super-optimal efficiency.

Motivated by developing faster solution algorithms than Algorithms 1-2 and by methods using narrow band region-based active contours, localised models amenable to fast solution are proposed in this section respectively for the BC model [12] and the RC model [111] . It is followed by developing the corresponding multilevel algorithms to solve them. As expected, the complexity of the new models will be directly linked to the length of segmented objects at each iteration; at the discrete level, this length is usually $O(\sqrt{N})$. Our use of narrow band regions is fundamentally different from active contours in that we apply the idea to a model, not just to a numerical procedure.

The key notation used below is the following, as shown in Figure 4.3. Given a level

set function ϕ (intended to represent Ω_1), a local function b defined by

$$b\left(\phi\left(x,y\right),\gamma\right)=H\left(\phi\left(x,y\right)+\gamma\right)\left(1-H(\phi(x,y)-\gamma)\right) \tag{4.26}$$

characterises the narrow band region domain $\Omega_\gamma=\Omega_1\left(\gamma\right)\cup\Gamma\cup\Omega_2\left(\gamma\right)$ surrounding the boundary Γ, with $\Omega_1\left(\gamma\right)$ and $\Omega_2\left(\gamma\right)$ denoting the γ band inside and outside region from Γ respectively. Similar notation is also used by [96, 152]. Note $b=1$ inside Ω_γ and 0 outside, and similarly $b\left(\phi\left(x,y\right),\gamma\right)H\left(\phi\right)=1$ inside $\Omega_1\left(\gamma\right)$ and 0 outside i.e. $b\left(\phi\left(x,y\right),\gamma\right)\left(1-H\left(\phi\right)\right)=1$ inside $\Omega_2\left(\gamma\right)$ and 0 outside. Further after discretisation, we define the notation for the set falling into the γ-band where $b=1$:

$$B(\phi)=\left\{(i,j)\ \middle|\ -\gamma\leq\phi_{i,j}\leq\gamma \text{ i.e. } \phi\left(x,y\right)+\gamma>0 \text{ and } \phi(x,y)-\gamma<0\right\}. \tag{4.27}$$

We propose a localised version of the BC model [12] by the following

$$\min_{\Gamma,c_1,c_2}\left\{F_{BL}\left(\Gamma,c_1,c_2\right)=\mu\int_\Gamma dgds+F_{BL}^\gamma\left(\Gamma,c_1,c_2\right)\right\} \tag{4.28}$$

where the refinement is seen in

$$F_{BL}^\gamma\left(\Gamma,c_1,c_2\right)=\lambda_1\int_{\Omega_1(\gamma)}\left(z-c_1\right)^2dxdy+\lambda_2\int_{\Omega_2(\gamma)}\left(z-c_2\right)^2dxdy.$$

In level set formation,

$$\begin{aligned}\min_{\phi,c_1,c_2}F_{BL}\left(\phi,c_1,c_2\right)=\mu\int_\Omega dg\sqrt{\left|\nabla H\left(\phi\right)\right|^2+\beta}\,dxdy+\lambda_1\int_\Omega\left(z-c_1\right)^2b\left(\phi,\gamma\right)H\left(\phi\right)dxdy\\ +\lambda_2\int_\Omega\left(z-c_2\right)^2b\left(\phi,\gamma\right)\left(1-H\left(\phi\right)\right)\,dxdy.\end{aligned} \tag{4.29}$$

Next, we propose a localised RC model of the form

$$\begin{aligned}&\min_{\phi,c_1,c_2}F_{RL}\left(\phi,c_1,c_2\right)=\mu\int_\Omega g\left(\left|\nabla z\left(x,y\right)\right|\right)\sqrt{\left|\nabla H\left(\phi\right)\right|^2+\beta}\,dxdy+\\ &\lambda_1\int_\Omega\left(z-c_1\right)^2b\left(\phi,\gamma\right)H\left(\phi\right)\,dxdy+\lambda_2\int_\Omega\left(z-c_2\right)^2b\left(\phi,\gamma\right)\left(1-H\left(\phi\right)\right)dxdy\\ &+\nu\Big(\int_\Omega b\left(\phi,\gamma\right)H\left(\phi\right)\,dxdy-\mathrm{A}_1\Big)^2+\nu\Big(\int_\Omega b\left(\phi,\gamma\right)\left(1-H\left(\phi\right)\right)\,dxdy-\mathrm{A}_2\Big)^2.\end{aligned} \tag{4.30}$$

4.5 Multilevel algorithms for localised segmentation models

We now show how to adapt the above Algorithms 1-2 to the new formulations (4.29) and (4.30).

Multilevel algorithm for the localised BC model. Discretise functional (4.29) as

$$\begin{aligned} F_{BL}\left(\phi, c_1, c_2\right) = \quad & \bar{\mu} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} G_{i,j} \sqrt{\left(H_{i,j} - H_{i,j+1}\right)^2 + \left(H_{i,j} - H_{i+1,j}\right)^2 + \beta} \\ & + \lambda_1 \sum_{i=1}^{n} \sum_{j=1}^{n} \left(z_{i,j} - c_1\right)^2 H_{i,j} b_{i,j} \\ & + \lambda_2 \sum_{i=1}^{n} \sum_{j=1}^{n} \left(z_{i,j} - c_2\right)^2 \left(1 - H_{i,j}\right) b_{i,j} \end{aligned} \tag{4.31}$$

where $G = dg,\ G_{i,j} = G(x_i, y_j),\ (i,j) \in B(\phi)$. Minimisation of (4.31) by the coordinate descent method on the finest level 1 leads to the following local minimisation for only $(i,j) \in B(\phi^{(m)})$:

$$\begin{aligned} F_{BL}^{loc}(\phi_{i,j}, c_1, c_2) = & \bar{\mu} \Big[G_{i,j} \sqrt{\left(H_{i,j} - H_{i+1,j}^{(m)}\right)^2 + \left(H_{i,j} - H_{i,j+1}^{(m)}\right)^2 + \beta} \\ & + G_{i-1,j} \sqrt{\left(H_{i,j} - H_{i-1,j}^{(m)}\right)^2 + \left(H_{i-1,j}^{(m)} - H_{i-1,j+1}^{(m)}\right)^2 + \beta} \\ & + G_{i,j-1} \sqrt{\left(H_{i,j} - H_{i,j-1}^{(m)}\right)^2 + \left(H_{i,j-1}^{(m)} - H_{i+1,j-1}^{(m)}\right)^2 + \beta} \Big] \\ & + \lambda_1 (z_{i,j} - c_1)^2 H_{i,j} b_{i,j} + \lambda_2 (z_{i,j} - c_2)^2 \left(1 - H_{i,j}\right) b_{i,j} \end{aligned} \tag{4.32}$$

where $b_{i,j} = 1$ if $(i,j) \in B(\phi^{(m)})$ else $b_{i,j} = 0$.

Further, the multilevel method for the localised BC model (4.29) at a general level for updating block $[k_1, k_2] \times [\ell_1, \ell_2]$ amounts to minimise the following local functional

$$\begin{aligned} F_{BC2}\left(c_{i,j}\right) = & \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell} \sqrt{\left(c_{i,j} - h_{k_1-1,\ell}\right)^2 + \upsilon_{k_1-1,\ell}^2 + \beta} \\ & + \bar{\mu} \sum_{k=k_1}^{k_2-1} G_{k,\ell_2} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_2}\right)^2 + h_{k,\ell_2}^2 + \beta} \\ & + \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2,\ell} \sqrt{\left(c_{i,j} - h_{k_2,\ell}\right)^2 + \upsilon_{k_2,\ell}^2 + \beta} \\ & + \bar{\mu} \sum_{k=k_1}^{k_2} G_{k,\ell_1-1} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_1-1}\right)^2 + h_{k,\ell_1-1}^2 + \beta} \\ & + \bar{\mu} \sqrt{2} G_{k_2,\ell_2} \sqrt{\left(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2}\right)^2 + \bar{h}_{k_2,\ell_2}^2 + \beta/2} \\ & + \lambda_1 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell) \in B(\tilde{\phi})} H(\tilde{\phi}_{k,\ell} + c_{i,j}) (z_{k,\ell} - c_1)^2 b(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma) \\ & + \lambda_2 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell) \in B(\tilde{\phi})} \big(1 - H(\tilde{\phi}_{k,\ell} + c_{i,j})\big) (z_{k,\ell} - c_2)^2 b(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma) \end{aligned} \tag{4.33}$$

Algorithm 3 BC2 – Multilevel algorithm for the new local BC model

- Input γ and the other quantities as in Algorithm 1
- Apply Algorithm 1 to new functionals from replacing

$$\min_{\phi_{i,j}} F^{loc}_{BC}(\phi_{i,j}, c_1, c_2) \quad \text{on the finest level by} \quad \min_{\phi_{i,j}} F^{loc}_{BL}(\phi_{i,j}, c_1, c_2)$$

$$\min_{c_{i,j}} F_{BC1}(c_{i,j}) \quad \text{on coarse levels by} \quad \min_{c_{i,j}} F_{BC2}(c_{i,j})$$

All other steps are identical.

similar to Algorithm 1, where $(i,j) \in B(\tilde{\phi})$. The difference is that $\tilde{\phi} := \tilde{\phi} + c_{i,j}$ only needs updating if the set $[k_1, k_2] \times [\ell_1, \ell_2] \ \cap \ B(\tilde{\phi})$ is non-empty. We will use the term **BC2** to refer the multilevel Algorithm 3.

Multilevel algorithm for the localised RC model. Functional (4.30) is discretised as

$$\begin{aligned} F_{RL}(\phi, c_1, c_2) = \ & \bar{\mu} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} g_{i,j} \sqrt{(H_{i,j} - H_{i,j+1})^2 + (H_{i,j} - H_{i+1,j})^2 + \beta} \\ & + \lambda_1 \sum_{i=1}^{n} \sum_{j=1}^{n} (z_{i,j} - c_1)^2 H_{i,j} b_{i,j} + \lambda_2 \sum_{i=1}^{n} \sum_{j=1}^{n} (z_{i,j} - c_2)^2 (1 - H_{i,j}) b_{i,j} \\ & + \nu \Big(-\mathrm{A}_1 + \sum_{i=1}^{n} \sum_{j=1}^{n} H_{i,j} b_{i,j} \Big)^2 + \nu \Big(-\mathrm{A}_2 + \sum_{i=1}^{n} \sum_{j=1}^{n} (1 - H_{i,j}) b_{i,j} \Big)^2. \end{aligned} \tag{4.34}$$

Further at a general level, whenever a block $[k_1, k_2] \times [\ell_1, \ell_2]$ overlaps with $B(\tilde{\phi})$ (i.e.

Algorithm 4 RC2 – Multilevel algorithm for the new and local RC model

- Input γ and the other quantities as in Algorithm 2
- Apply Algorithm 2 to new functionals from replacing

$$\min_{\phi_{i,j}} F^{loc}_{RC}(\phi_{i,j}, c_1, c_2) \quad \text{on the finest level by} \quad \min_{\phi_{i,j}} F^{loc}_{RL}(\phi_{i,j}, c_1, c_2)$$

$$\min_{c_{i,j}} F_{RC1}(c_{i,j}) \quad \text{on coarse levels by} \quad \min_{c_{i,j}} F_{RC2}(c_{i,j})$$

All other steps are identical.

the set $[k_1, k_2] \times [\ell_1, \ell_2] \cap B(\tilde{\phi})$ is non-empty), the multilevel method minimises

$$\begin{aligned}
F_{RC2}(c_{i,j}) =& \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{(c_{i,j} - h_{k_1-1,\ell})^2 + \upsilon^2_{k_1-1,\ell} + \beta} \\
&+ \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{(c_{i,j} - \upsilon_{k,\ell_2})^2 + h^2_{k,\ell_2} + \beta} \\
&+ \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{(c_{i,j} - h_{k_2,\ell})^2 + \upsilon^2_{k_2,\ell} + \beta} \\
&+ \bar{\mu}\sqrt{2} g_{k_2,\ell_2} \sqrt{(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2})^2 + \bar{h}^2_{k_2,\ell_2} + \frac{\beta}{2}} \\
&+ \bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{(c_{i,j} - \upsilon_{k,\ell_1-1})^2 + h^2_{k,\ell_1-1} + \beta} \\
&+ \lambda_1 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} b\left(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma\right) H(\tilde{\phi}_{k,\ell} + c_{i,j}) \, (z_{k,\ell} - c_1)^2 \\
&+ \lambda_2 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} \left(1 - H(\tilde{\phi}_{k,\ell} + c_{i,j})\right) b\left(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma\right) (z_{k,\ell} - c_2)^2 \\
&+ \nu \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} \left(-\mathrm{A}_1 + b\left(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma\right) H\left(\tilde{\phi}_{k,\ell} + c_{i,j}\right)\right)^2 \\
&+ \nu \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} \left(-\mathrm{A}_2 + \left(1 - H\left(\tilde{\phi}_{k,\ell} + c_{i,j}\right)\right) b\left(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma\right)\right)^2
\end{aligned} \tag{4.35}$$

and then updates $\tilde{\phi}$ by $\tilde{\phi} + c_{i,j}$. We will refer this Algorithm 4 as **RC2**.

For a single object extraction, Algorithms 3 - 4 have a complexity of $O(\gamma n \log N) = O(\sqrt{N} \log N)$ where $\log N$ refers to the number of levels. However they are only applicable to our selective models; for global models such as the CV model, the band idea promotes local minimisers and is hence not useful.

4.6 Numerical experiments

In order to demonstrate the strengths and limitations of the proposed multilevel method for both the original and the localised segmentation models, we performed several experiments. The main algorithms to be compared are

Name	Algorithm		Description
BC0	Old	:	The AOS algorithm [12] for the original BC model [12].
BCP	Old	:	The Pyramid scheme for BC0.
BC1	New	:	The multilevel Algorithm 1 for the BC model.
BC2	New	:	The multilevel Algorithm 3 for the localised BC model.
RC0	Old	:	The AOS algorithm [111] for the original RC model [111].
RCP	Old	:	The Pyramid scheme for RC0.
RC1	New	:	The multilevel Algorithm 2 for the RC model.
RC2	New	:	The multilevel Algorithm 4 for the localised RC model.

Our aims of the tests are
i) to verify numerically the efficiency as n increases – is an algorithm faster or slower than or of the same magnitude as $O(N \log N)$ where $N = n^2$;
ii) to compare the quality, we use the so-called the Jaccard similarity coefficient (JSC) and Dice similarity coefficient (DSC):

$$JSC = \frac{|S_n \cap S_*|}{|S_n \cup S_*|}, \qquad DSC = \frac{2\,|S_n \cap S_*|}{|S_n| + |S_*|}$$

where S_n is the set of the segmented domain Ω_1 and S_* is the true set of Ω_1. The similarity functions return values in the range $[0, 1]$. The value 1 indicates perfect segmentation quality while the value 0 indicates poor quality.

The test images used in this paper are listed in Figure 4.4. They are four images used which include 3 real medical images and 1 synthetic image. The synthetic image is created manually in Matlab in binary format so that the true set solution S_* is known to aid computing JSC and DSC. The markers set also are shown in Figure 4.4. The initial contour is defined by the markers set. We remark that for an image of size $n \times n$ the number of unknowns is $N = n^2$ which means that for $n = 256,\ 512,\ 1024,\ 2048$, the respective number of unknowns is $N = 65536,\ 262144,\ 1048576,\ 4194304$; we are solving large scale problems. Our algorithms are implemented in MATLAB R2017a on a computer with Intel Core i7 processor, CPU 3.60GHz, 16 GB RAM CPU. All the programs are stopped when $tol = 10^{-4}$ or when the maximum number of iterations, $maxit = 1500$ is reached.

4.6.1 Comparison of BC2 with BC0, BC1, and BCP

In the following experiments, we take the parameters $\lambda_1 = \lambda_2 = 1$, $\alpha = 0.01$, $\beta = 10^{-4}$ and $\kappa = 4$. Through the experiments it was observed that the parameters ε and η can be in a range between $\varepsilon = 1/n$ to 1 and $\eta = 10^{-3}$ and 10^2.

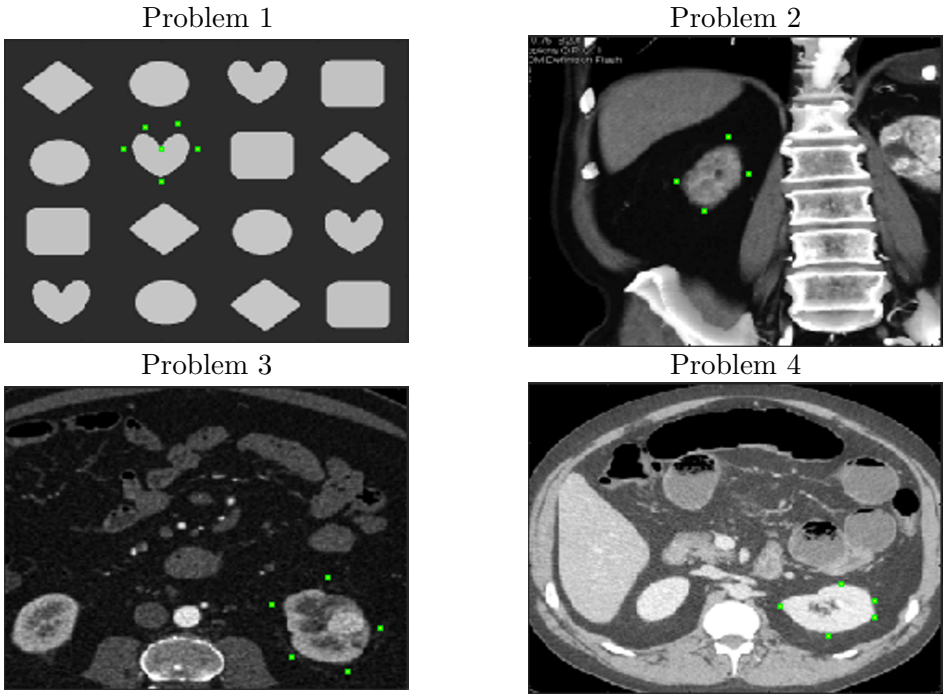


Figure 4.4: Test images with the markers set.

Table 4.1: Comparison of computation time (in seconds) of BC1 with BC2 for Problems 1-3. Note BC2 is about 2 times faster than BC1.

Problem	BC1	BC2
1	12.1	8.4
2	11.7	6.8
3	11.9	9.3

First, we compare BC1 and BC2 using test Problems 1-3. All the images are of size 256×256. We take $\bar{\mu} = 0.05n^2$ (Problem 1) and $\bar{\mu} = 0.1n^2$ (Problems $2-3$). For BC2, γ is between 30 to 100.

By visual evaluation, Figure 4.5(a) and 4.5(b) show the successful selective segmentation results by BC1 and BC2 respectively for capturing one object in Problems 1-3. We see that that the results from BC1 are quite similar to BC2. The times needed by BC1 and BC2 to complete the selective segmentation task are tabulated in Table 4.1 where we can observe that BC2 is about 2 times faster than BC1.

Second, against BC2, we test BC0 that is based on additive operator splitting (AOS) [12], the pyramid scheme BCP based on BC0 and BC1. For this purpose, we segment Problem 1 with different resolutions using $\mu = \bar{\mu} = 0.05n^2$. The segmentation results for image size 1024×1024 are shown in Figure 4.6 and the results for all sizes in terms of quality and computation time needed to complete the segmentation tasks are presented in Table 4.2. Columns 5 (ratios of the CPU times) show that BC0, BCP, and BC1 are of complexity $O(N \log N)$ while BC2 of the 'super'-optimal efficiency $O(\sqrt{N} \log N)$.

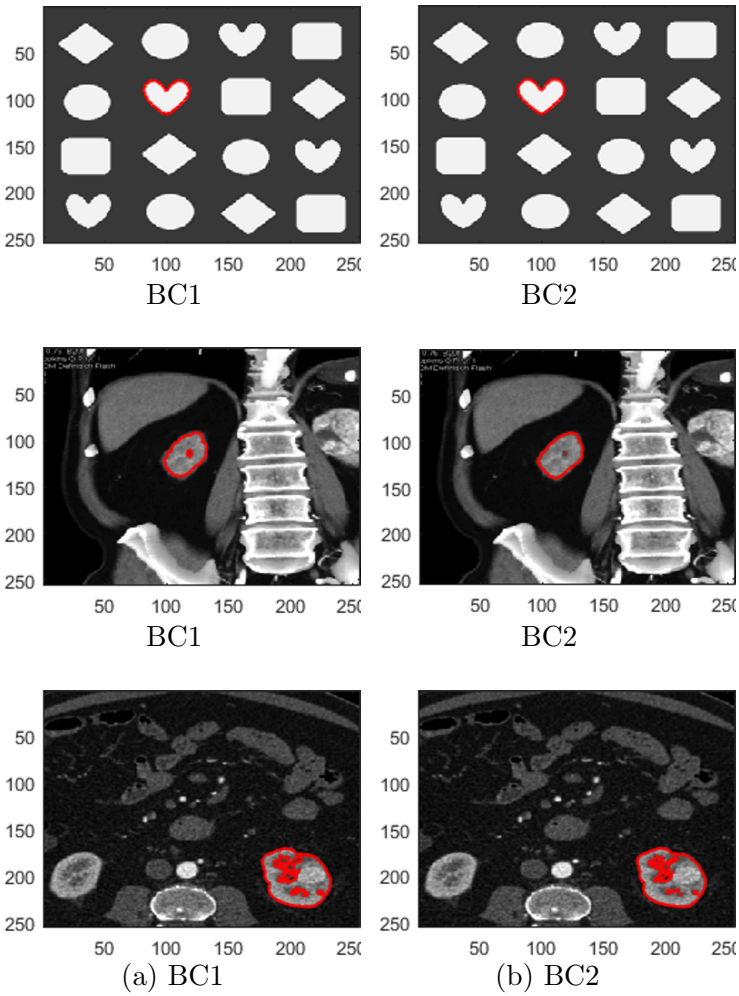


Figure 4.5: Segmentation of Problems 1-3: Column (a) BC1 and (b) BC2.

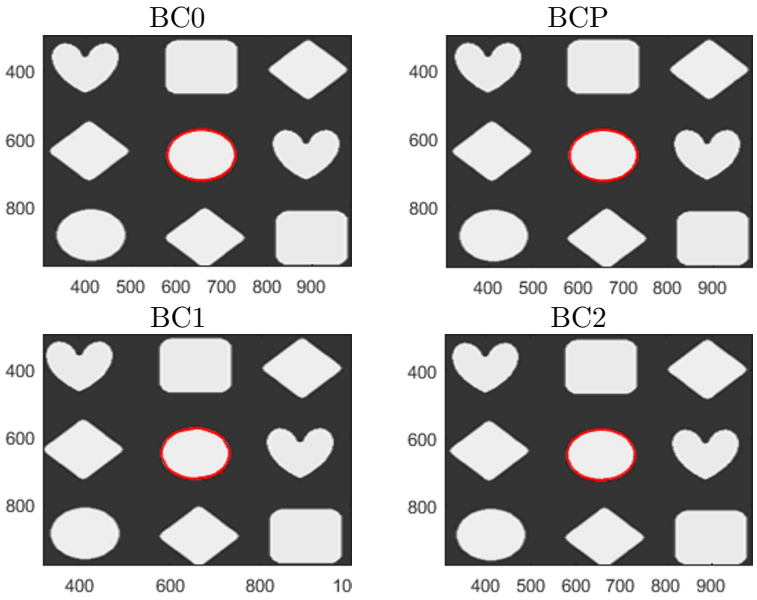


Figure 4.6: Segmentation of Problem 1 of size 1024×1024 for BC0, BCP, BC1, and BC2.

Table 4.2: Comparison of computation time (in seconds) and segmentation quality of BC0, BCP, and BC1 with our BC2 for Problem 1. The ratio close to 4.4 for time indicates $O(N \log N)$ speed while a ratio of 2.2 indicates $O(\sqrt{N} \log N)$ "super-optimal" speed, where the number of unknowns $N = n^2$. Here and later, '**' means taking too long to run (> 24 hours).

Algorithm	Size $n \times n$	Number of iteration (outer)	Time t_n	$\frac{t_n}{t_{n-1}}$	JSC	DSC
BC0	256×256	1293	227.8		1.0	1.0
	512×512	1276	898.5	3.9	1.0	1.0
	1024×1024	1234	4095.5	4.6	1.0	1.0
	2048×2048	**	**	-	-	-
BCP	256×256	4	61.0		1.0	1.0
	512×512	2	180.0	3.0	1.0	1.0
	1024×1024	2	812.3	4.5	1.0	1.0
	2048×2048	2	3994.0	4.9	1.0	1.0
BC1	256×256	2	11.6		1.0	1.0
	512×512	2	43.7	3.8	1.0	1.0
	1024×1024	2	173.2	4.0	1.0	1.0
	2048×2048	2	736.9	4.3	1.0	1.0
BC2	256×256	2	10.5		1.0	1.0
	512×512	2	21.6	2.1	1.0	1.0
	1024×1024	2	42.5	2.0	1.0	1.0
	2048×2048	2	80.5	1.9	1.0	1.0

Clearly BC0 (the AOS method for the BC model) provides an effective acceleration for images of moderate size $n \leq 256$. Significant improvement can be seen by BCP which shows that the pyramid method together with AOS is better than BC0. However, we can see that our BC1 and BC2 are faster than BC0 and BCP, while BC2 is faster than the other 3 algorithms. The computation time differences between BC2 with other 3 algorithms become significant as the image size increases to $n \geq 512$. The BC0 result with " ** " indicates that a very long time is taken to complete the segmentation task (> 24 hours). One can see for example BC0 needs almost 100 times more time compared to BC2 to complete segmentation in case of 1024×1024. We also see that all algorithms provide high segmentation quality, from JSC and DSC values.

4.6.2 Comparison of RC2 with RC0, RC1, and RCP

In the following experiments, we fixed the parameters $\lambda_1 = \lambda_2 = 1$, $\alpha = 0.01$ and $\beta = 10^{-4}$. Through the experiments it was observed that the parameters ν, ε and η can be in a range between $\nu = 0.001$ to 0.01 $\varepsilon = 1/n$ to 1 and $\eta = 10^{-3}$ to 10^{-2}.

We first compare RC1 and RC2 using Problems 1-3. All the images are of size 256×256. We take $\bar{\mu} = 0.05n^2$ (Problem 1) and $\mu = \bar{\mu} = 0.1n^2$ (Problems 2-3). For RC2, γ is between 30 to 100. Figure 4.7(a) and 4.7(b) show the successful selective segmentation results of RC1 and RC2 respectively for capturing one object for Problems 1-3.

Table 4.3: Comparison of computation time (in seconds) and segmentation quality of RC0, RCP, and RC1 with RC2 for Problem 1. Again, the ratio close to 4.4 for time indicates $O(N \log N)$ speed while a ratio of 2.2 indicates $O(\sqrt{N} \log N)$ "super-optimal" speed, where the number of unknowns $N = n^2$.

Algorithm	Size $n \times n$	Number of iteration (outer)	Time t_n	$\frac{t_n}{t_{n-1}}$	JSC	DSC
RC0	256×256	1500	260.5		1.0	1.0
	512×512	1385	975.0	3.7	1.0	1.0
	1024×1024	1404	4735.0	4.9	1.0	1.0
	2048×2048	**	**	-	-	-
RCP	256×256	4	62.5		1.0	1.0
	512×512	2	187.2	3.0	1.0	1.0
	1024×1024	2	822.3	4.4	1.0	1.0
	2048×2048	2	3996.3	4.9	1.0	1.0
RC1	256×256	2	13.0		1.0	1.0
	512×512	2	48.4	3.7	1.0	1.0
	1024×1024	2	189.9	3.9	1.0	1.0
	2048×2048	2	819.0	4.3	1.0	1.0
RC2	256×256	2	11.5		1.0	1.0
	512×512	2	24.0	2.1	1.0	1.0
	1024×1024	2	46.9	2.0	1.0	1.0
	2048×2048	2	87.6	1.9	1.0	1.0

We then compare RC2 with RC0, RCP, and RC1 using Problem 1. Here $\mu = \bar{\mu} = 0.05n^2$ for all algorithms. The segmentation results for image size 1024×1024 shown in Figure 4.8 and the quality measures and the computation time presented in Table 4.3 show that RC2 can be 100 times faster than RC0, 17 times faster than RCP and 4 times faster than RC1 for the case of 1024×1024: In particular, ratios of the CPU times verify that RC0, RCP, and RC1 are of complexity $O(N \log N)$ while RC2 of the 'super'-optimal efficiency $O(\sqrt{N} \log N)$. Furthermore, the RC0 result with " ** " indicates that too much time is taken to complete the segmentation task (> 24 hours). The high values of JSC and DSC show that RC0, RCP, RC1, and RC2 provide high segmentation quality.

For the benefit of readers, in Figure 4.9, we demonstrate a convergent plot based on Tables 4.2 and 4.3 of our proposed multilevel based models (BC2 and RC2) in segmenting Problem 1 of size 2048×2048. One can see that the models are fast, converging to *tol* in 2 iterations that is before the prescribed *maxit*.

Furthermore, we have extended the number of iterations for BC2 and RC2 up to 6 iterations and plot the residual history in the same Figure 4.9. We can observe that BC2 and RC2 keep converging.

4.6.3 Sensitivity tests on the algorithmic parameters

Sensitivity is a major issue that has to be addressed and tested below. We shall pay particular attention to the regularising parameter β that was known to be sensitive to

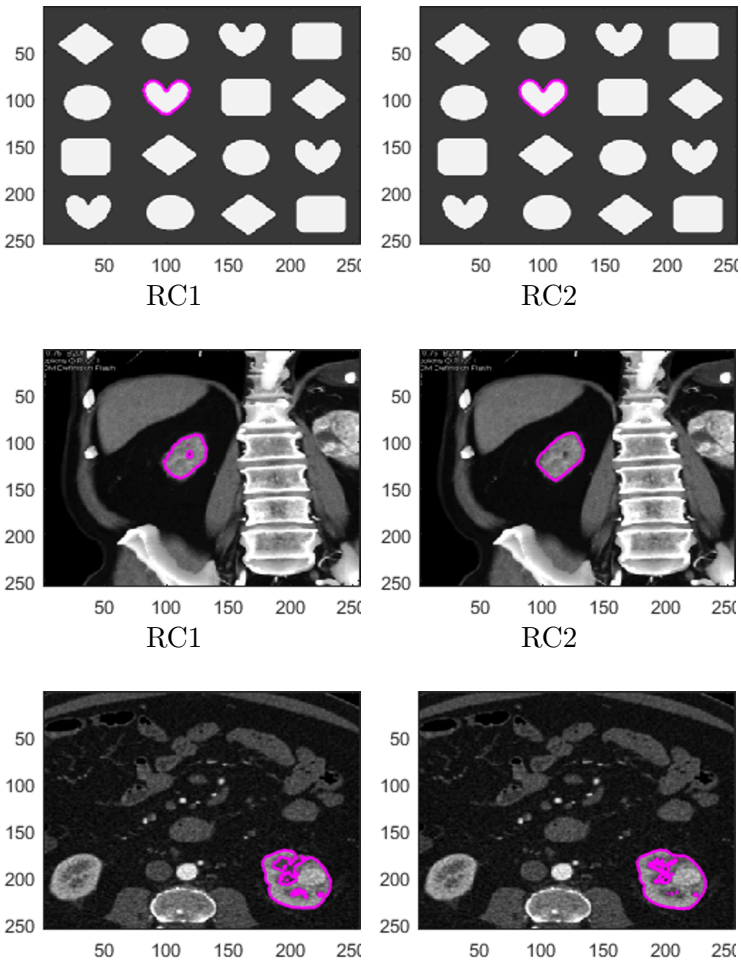


Figure 4.7: Segmentation of Problems 1-3. (a) and (b) show the segmentation using RC1 and RC2 respectively.

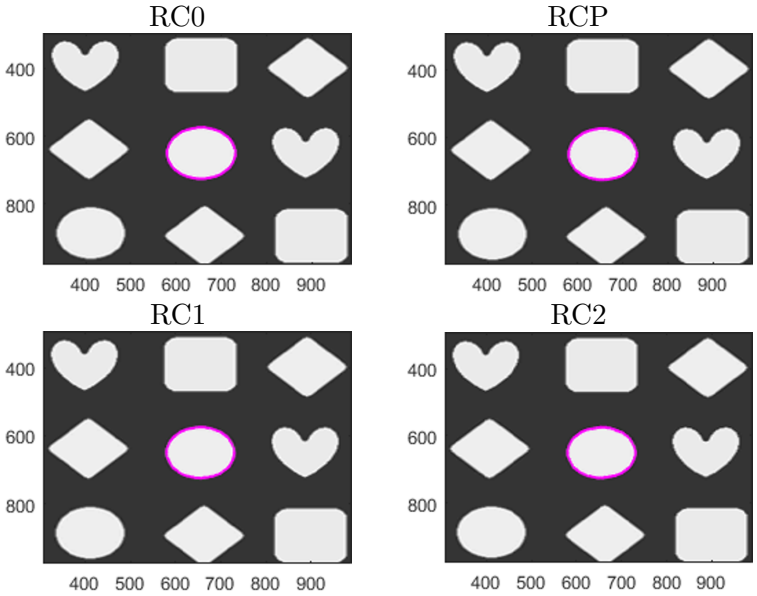


Figure 4.8: Segmentation of Problem 1 of size 1024×1024 for RC0, RCP, RC1, and RC2. For the same segmentation result, RC2 can be 100 times faster than RC0, 17 times faster than RCP and 4 times faster than RC1; see Table 4.3.

Table 4.4: Dependence of BC2 and RC2 on t for heart shape in Problem 1 (Figure 4.4).

Algorithm	t : inner iteration	Number of iteration (outer)	CPU	JSC	DSC
BC2	1	2	8.1	1.0	1.0
	2	6	22.4	1.0	1.0
	3	7	26.7	0.9	1.0
RC2	1	2	8.8	1.0	1.0
	2	9	35.3	0.9	1.0
	3	7	28.7	0.9	1.0

convergence of a geometric multigrid method [10]; it turns out that our Algorithms 1-4 are more advantageous as they are not very sensitive to β.

Tests on parameter t. The inner iteration t indicates the number of iterations needed to solve the minimisation problem in each level. We test several numbers of t required by BC2 and RC2 to segment heart shape in Problem 1 and record the outer iteration needed to achieve *tol*, the CPU time and the quality of segmentation. The results are tabulated in Table 4.4.

We can see that BC2 and RC2 work in efficiently and effectively using only 1 inner iteration i.e $t = 1$. As we increase t, the quality of segmentation for BC2 and RC2 reduce and need more CPU time and outer iteration as well.

Tests on parameter γ. The band width parameter γ is an important parameter to be tested. Its size determines how local the resulting segmentation will be. Below we will demonstrate the effect of applying different values of γ to BC2 and RC2. We aim to segment an organ in Problem 4 by for BC2 and RC2 with varying γ and the results are presented in Figure 4.10. Columns 2 and 3 of Figure 4.10 show the results using 3 γ values (more spread out) for BC2 and RC2. Clearly, unless the value is too small (that results in an incorrect segmentation), in general, both BC2 and RC2 are not much sensitive to γ choice.

Tests on parameter β. Finally, we examine the sensitivity of BC2 and RC2 on this important parameter β. 7 different β are tested: $\beta = 1,\ 10^{-1},\ 10^{-2},\ 10^{-4},\ 10^{-6},\ 10^{-8}$ and 10^{-10} in segmenting the heart shape in Problem 1. For a quantitative analysis, we evaluate the energy value $F_{BL}(\phi, c_1, c_2)$ in equation (4.31), $F_{RL}(\phi, c_1, c_2)$ in equation (4.34) and the indexes JSC and DSC. The values of $F_{BL}(\phi, c_1, c_2)$, $F_{RL}(\phi, c_1, c_2)$, JSC and DSC are tabulated in Table 4.5. One can see that as β decreases, the functional $F_{BL}(\phi, c_1, c_2)$ and $F_{RL}(\phi, c_1, c_2)$ gets closer to each other. The segmentation quality measured by JSC and DSC increases as β decreases. This finding shows that BC2 and RC2 are only sensitive to (unrealistic) large β but less to a very small β. In separate experiments, we found that BC2 and RC2 algorithms are not much sensitive to η, α, ε, and ν (involve in RC2 only), although there exist choices what give the optimal quality of segmentation.

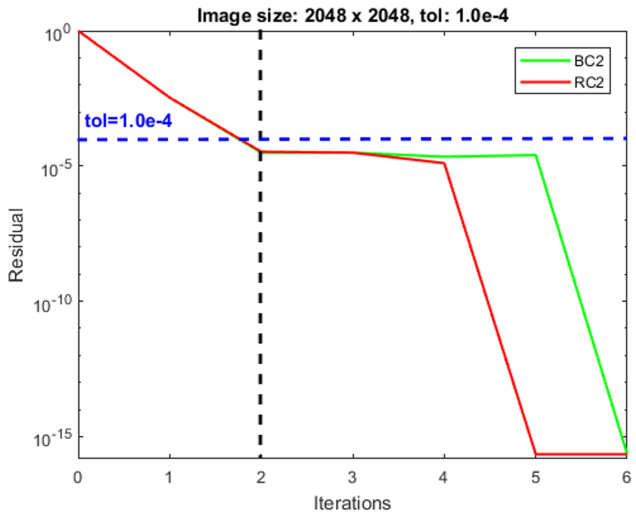
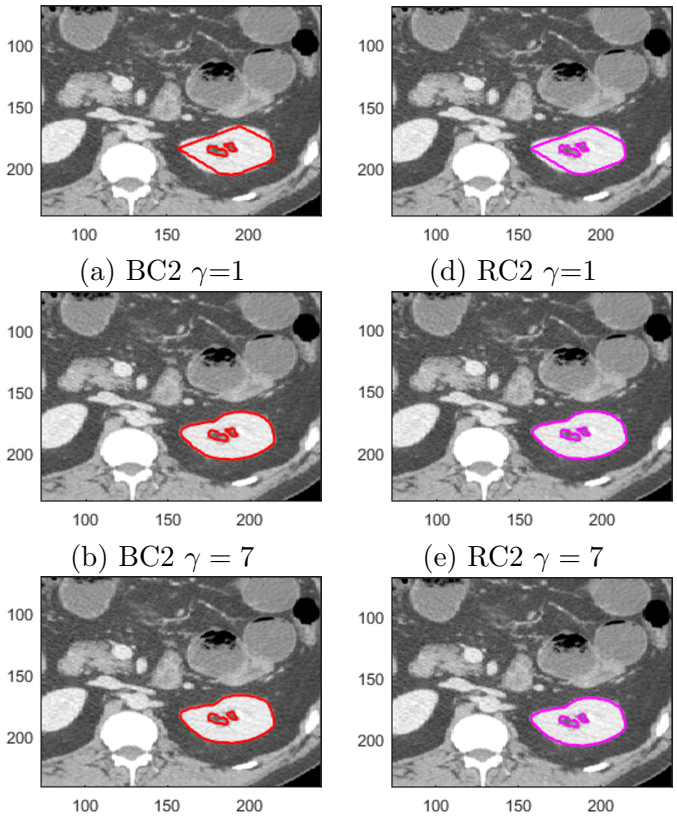


Figure 4.9: The number of iterations needed by BC2 and RC2 to achieve a set *tol* (residual) in segmenting an image of size 2048×2048. With $tol = 10^{-4}$, BC2 and RC2 need 2 iterations. The extension up to 6 iterations shows that residuals of BC2 and RC2 keep reducing.



(a) BC2 γ=1 (d) RC2 γ=1

(b) BC2 $\gamma = 7$ (e) RC2 $\gamma = 7$

(c) BC2 $\gamma = 100$ (f) RC2 $\gamma = 100$

Figure 4.10: Dependence of algorithms BC2, RC2 on parameter γ for Problem 4.

Table 4.5: Dependence of our new BC2 and RC2 on β for heart shape in Problem 1 (Figure 4.4).

β	BC2			RC2		
	$F_{BL}(\phi, c_1, c_2)$	JSC	DSC	$F_{RL}(\phi, c_1, c_2)$	JSC	DSC
1	2.461759e+09	0.6	0.7	5.177135e+10	0.6	0.7
10-1	2.258762e+09	0.9	1.0	5.168056e+10	0.9	1.0
10-2	2.197002e+09	1.0	1.0	5.164663e+10	1.0	1.0
10-4	2.178939e+09	1.0	1.0	5.163375e+10	1.0	1.0
10-6	2.177950e+09	1.0	1.0	5.163266e+10	1.0	1.0
10-8	2.176280e+09	1.0	1.0	5.163252e+10	1.0	1.0
10-10	2.175254e+09	1.0	1.0	5.163243e+10	1.0	1.0

4.7 Summary

In this chapter, we presented an optimisation based multilevel method to solve two variational and selective segmentation models (BC and RC), though the idea is applicable to other global and variational models. Firstly, we presented 2 algorithms (BC1, RC1) for solving the respective models with each algorithm having the expected optimal complexity of $O(N \log N)$ for segmentation of an image of size $n \times n$ or $N = n^2$ unknowns (pixels). These algorithms can be adapted to solve other segmentation models. Secondly, we reformulated the models so that they became localised versions that operate within a banded region of an active level set contour, and consequently obtained 2 further algorithms (BC2, RC2) with each algorithm having the 'super'-optimal complexity of approximately $O(\sqrt{N} \log N)$ depending on the objects to be segmented. These algorithms are only applicable to our selective segmentation models. Numerical experiments have verified the complexity claims, and comparisons with related algorithms (BC0, BCP, RC0, RCP for the standard models) show that the new algorithms are many times faster than BC0, BCP, RC0, RCP, in achieving comparable quality of segmentation. All the aboves models are nonconvex. In the next chapter, we will address convexified selective variational models and explore their advantages.

Chapter 5

A Reformulated Convex and Selective Variational Image Segmentation Model and its Fast Multilevel Algorithm

In the previous chapter, we have proposed a fast solver for a class of selective models. However, they would find local minimisers (sensitive to initialisation) because nonconvex minimisation functionals are involved. Recently, Spencer-Chen [129] has successfully proposed a convex selective variational image segmentation model (named CDSS), allowing a global minimiser to be found independently of initialisation. However, their algorithm is sensitive to the regularisation parameter μ and the area parameter θ due to nonlinearity in the functional and additionally it is only effective for images of moderate size. In this chapter, a stabilised variant of CDSS model through primal-dual formulation is proposed and an optimisation based multilevel algorithm for the new model is introduced. Numerical results show that the new model is less sensitive to parameter μ and θ compared to the original CDSS model and the multilevel algorithm produces quality segmentation in optimal computational time.

5.1 Introduction

Selective image segmentation aims to extract one object of interest in an image based on some additional information of geometric constraints [65, 111, 129]. This task cannot be achieved by global segmentation. Some effective models are Badshah-Chen [12] and Rada-Chen [111] which used a mixed edge based and region based ideas, and area constraints. Both models are nonconvex. A nonconvex selective variational image segmentation model, though effective in capturing a local minimiser, is sensitive to initialisation where the segmentation result relies heavily on user input.

While the above selective segmentation models are formulated based on geometric constraints in [65, 67], there are another way of defining the geometric constraints that

can be found in [103] where geometric points outside and inside a targeted object are given. Their model make use the Split Bregman method to speed up convergence. Although our paper based on geometric constraint defining in [65, 67], later, we shall compare our work with [103]. We called their model as NCZZ model.

In 2015, Spencer-Chen [129, 128] has successfully designed a Convex Distance Selective Segmentation model (named as CDSS). This variational model allows a global minimiser to be found independently of initialisation, given knowledge of $c_1,\ c_2$. The CDSS model [129] is challenging to solve due to its penalty function $\nu(u)$ being highly nonlinear. Consequently, the standard addition operator splitting method (AOS) is not adequate. An enhanced version of the AOS scheme was proposed in [129] by taking the approximation of $\nu'(u)$ which based on its linear part [129, 128]. Another factor that affects the [129] model is how to choose the combination values of the regularisation parameters μ and θ (other parameters can be fixed as suggested by [129, 128]). For a simple (synthetic) image, it is easy to get a suitable combination of parameter μ and θ which gives a good segmentation result. However, for other real life images, it is not trivial to determine a suitable combination of μ and θ simultaneously; our experiments show that high segmentation accuracy is given by the model in a small range of μ and θ and consequently the model is not ready for general use. Of course, it is known that an AOS method is not designed for processing large images.

We remark that a most recent convex selective variational image segmentation was introduced by Liu *et al.* [83]. in 2018. The work is based on [12, 23, 111]. We named their model as CMT model. Although our work is based on [129, 128], we shall compare our work with CMT model [83] later.

This chapter investigates both the robust modeling and fast solution issues by making two contributions. Firstly, we propose a better model than CDSS. In looking for possible improvement on the selective model CDSS, we are inspired by several works [20, 8, 9, 26, 35, 24] on non-selective segmentation. The key idea that we will employ in our new model is the primal-dual formulation which allows us to "ignore" the penalty function $\nu(u)$, otherwise creating problems of parameter sensitivity. We remark that similar use of the primal-dual idea can be found in D. Chen et al. [40] to solve a variant of Mumford-Shah model which handles the segmentation of medical images with intensity inhomogeneities and also in Moreno et al. [99] for solving a four phase model for segmentation of brain MRI images by active contours. Secondly, we propose a fast optimisation based multilevel method for solving the new model, which is applicable to the original CDSS [129], in order to achieve fast convergence especially for images with large size. We will consider the differentiable form of variational image segmentation models and develop the multilevel algorithm for the resulting models without using a "patch detection" idea. We are not aware of similar work done for segmentation models in the variational convex formulation.

The rest of this chapter is organised in the following way. In Section 5.2, we first briefly review the convex selective segmentation models. These models are NCZZ model [103], CDSS model [129], and CMT model [83]. In Section 5.3, we give our

new primal-dual formulation of the CDSS model and in Section 5.4 we present the optimisation based multilevel algorithm. We proposed a new variant of the multilevel algorithm in Section 5.5 and discuss their convergence in Section 5.6. In Section 5.7 we give some experimental results before concluding in Section 5.8.

The work presented in this chapter has been published by me as the main author together with my supervisor Prof. Ke Chen as the co-author in [74]. The new primal-dual formulation and its multilevel algorithm were developed and implemented in MATLAB by the author. The new variant of the multilevel algorithm and its convergence were jointly developed by us while the algorithm was implemented in MATLAB by the author.

5.2 Convex selective segmentation models

As discussed, there exist many variational segmentation models in the literature on global segmentation and few on selective image segmentation models in convex formulation. For the latter, we will review three segmentation models below that are directly related to this work.

5.2.1 NCZZ model

This interactive selective segmentation model is proposed by Nguyen et al. [103]. Given an image $z = z(x, y)$, this model has the energy functional

$$\begin{aligned} F_{NCZZ}(\phi) = \mu \int_{\Omega} g\left(|\nabla z(x,y)|\right) |\nabla \phi| \, d\Omega \\ + \int_{\Omega} \alpha \left(P_B(x,y) - P_F(x,y)\right) + (1-\alpha)\left(1 - 2P(x,y)\right) \phi \, d\Omega, \end{aligned} \tag{5.1}$$

where P_B and P_F are the normalized log-likelihoods that the pixel (x, y) is in the foreground and background respectively. The function $P(x, y)$ is the probability that pixel (x, y) belongs to the foreground and $\alpha, \phi \in [0, 1]$. In this model $g(s) = \frac{1}{1+\gamma s^2}$ is an edge detector function which helps to stop the evolving curve on the edge of the objects in an image. The strength of detection is adjusted by parameter γ. Another type of edge detector function considered is $g(s) = \beta g_c + (1 - \beta) g_e$ where g_c and g_e are the results of applying the edge detection to P_F and the original image, respectively and $\beta \in [0, 1]$. This model is good for many examples, see [111, 103], however fails when the boundary of the targeted object is non-smooth or has fine structure. As explained by the authors, these limitations of the model lie in its underlying assumption that the shape of the object is smooth and can be well described by the weighted shortest boundary length. In addition, the model is essentially a hard segmentation method where a pixel is assigned to exactly one class [103].

5.2.2 CMT model

Recently, a selective convex model was introduced by Liu et al. [83] which applies a weighting to the data fitting terms. The functional is given by

$$F_{CMT}(u) = \int_{\Omega} |\nabla u| \, d\Omega + \int_{\Omega} |\nabla u|^2 \, d\Omega + \int_{\Omega} \omega^2(x,y)|z-u|^2 \, d\Omega. \tag{5.2}$$

Here, $\omega(x,y) = 1 - d(x,y)g(|\nabla z|)$ where $d(x,y)$ is a distance function from marker set A. This model is based on Mumford-Shah model [102] and contains two stages. The first stage is to obtain a smooth approximation related to the Mumford-Shah model. The second stage is to use this smooth approximation and perform a thresholding procedure to obtain the object of interest.

5.2.3 Convex Distance Selective Segmentation model

Assume that an image $z = z(x,y)$ comprises of two regions of approximately piecewise constant intensities of distinct values (unknown) c_1 and c_2, separated by some (unknown) curve or contour Γ. Let the object to be detected be represented by the region Ω_1 with the value c_1 inside the curve Γ whereas outside Γ, in $\Omega_2 = \Omega \backslash \Omega_1$, the intensity of z is approximated with value c_2. In a level set formulation, the unknown curve Γ is represented by the zero level set of the Lipschitz function such that

$$\begin{aligned}
&\Gamma = \{(x,y) \in \Omega : \phi(x,y) = 0\}, \\
&\Omega_1 = \text{inside}(\Gamma) = \{(x,y) \in \Omega : \phi(x,y) > 0\}, \\
&\Omega_2 = \text{outside}(\Gamma) = \{(x,y) \in \Omega : \phi(x,y) < 0\}.
\end{aligned}$$

Let n_1 geometric constraints be given by a marker set

$$A = \{w_i = (x_i^*, y_i^*) \in \Omega, 1 \leq i \leq n_1\} \subset \Omega$$

where each point is near the object boundary Γ, not necessarily on it [111, 152]. The selective segmentation idea tries to detect the boundary of a single object among all homogeneity intensity objects in Ω close to A; here n_1 (≥ 3). The geometrical points in A define an initial polygonal contour and guide its evolution towards Γ [152].

The Distance Selective Segmentation (DSS) model [129] was proposed by Spencer and Chen [129] in 2015. The formulation is based on the special case of the piecewise constant Mumford-Shah functional [102] where it is restricted to only two phase (i.e. constants), representing the foreground and the background of the given image $z(x,y)$.

Using the set A, construct a polygon Q that connects up the markers. Denote the function $P_d(x,y)$ as the Euclidean distance of each point $(x,y) \in \Omega$ from its nearest point $(x_p, y_p) \in Q$:

$$P_d(x,y) = \sqrt{(x-x_p)^2 + (y-y_p)^2} = \min_{q \in Q} \|(x,y) - (x_q, y_q)\|$$

and denote the regularised versions of a Heaviside function by

$$H_\varepsilon\left(\phi\left(x,y\right)\right)=\frac{1}{2}\left(1+\frac{2}{\pi}\arctan\left(\frac{\phi}{\varepsilon}\right)\right).$$

Then the DSS in a level set formulation is to minimise a cost function defined as follows

$$\begin{aligned}\min_{\phi,c_1,c_2} D\left(\phi,c_1,c_2\right)= & \ \mu\int_\Omega g\left(\left|\nabla z\right|\right)\left|\nabla H_\varepsilon(\phi)\right|d\Omega+\int_\Omega H_\varepsilon\left(\phi\right)\left(z-c_1\right)^2 d\Omega \\ & +\int_\Omega\left(1-H_\varepsilon\left(\phi\right)\right)\left(z-c_2\right)^2 d\Omega+\theta\int_\Omega H_\varepsilon\left(\phi\right)P_d\ d\Omega\end{aligned} \tag{5.3}$$

where μ and θ are nonnegative parameters. The addition of new distance fitting term is weighted by the area parameter θ. Here, if the parameter θ is too strong the final result will just be the polygon P which is undesirable.

The above model from (5.3) was relaxed to obtain a constrained Convex Distance Selective Segmentation (CDSS) model [129]. This was to make sure that the initialisation can be flexible. The CDSS was obtained by relaxing $H_\varepsilon\rightarrow u\in[0,1]$ to give:

$$\min_{0\leq u\leq 1} CDSS\left(u,c_1,c_2\right)=\mu\int_\Omega\left|\nabla u\right|_g d\Omega+\int_\Omega ru\ d\Omega+\theta\int_\Omega P_d u\ d\Omega \tag{5.4}$$

and further an unconstrained minimisation problem:

$$\min_u CDSS\left(u,c_1,c_2\right)=\mu\int_\Omega\left|\nabla u\right|_g\ d\Omega+\int_\Omega ru\ d\Omega+\theta\int_\Omega P_d u\ d\Omega+\alpha\int_\Omega\nu\left(u\right)\ d\Omega \tag{5.5}$$

where $r=\left(c_1-z\right)^2-\left(c_2-z\right)^2$ and $\left|\nabla u\right|_g=g\left(\left|\nabla z\right|\right)\left|\nabla u\right|$, $\nu\left(u\right)=\max\left\{0,2\left|u-\frac{1}{2}\right|-1\right\}$ is an exact (non-smooth) penalty term, provided that $\alpha>\frac{1}{2}\|r+\theta P_d\|_{L^\infty}$ (see also [34]). For fixed c_1, c_2, μ, θ, and $\kappa\in[0,1]$, the minimiser u of (5.4) is guaranteed to be a global minimiser defining the object by $\sum=\{(x,y):u\left(x,y\right)\geq\kappa\}$ [129, 34, 20]. The parameter κ is a threshold value and usually $\kappa=0.5$.

In order to compute the associated Euler Lagrange equation for u they introduce the regularised version of $\nu\left(u\right)$:

$$\nu\left(u\right)=\left[\sqrt{\left(2u-1\right)^2+\varepsilon}-1\right]H\big(\sqrt{\left(2u-1\right)^2+\varepsilon}-1\big),\quad H\left(x\right)=\frac{1}{2}+\frac{1}{\pi}\arctan\left(\frac{x}{\varepsilon}\right).$$

Consequently, the Euler Lagrange equation for u in equation (5.5) is the following

$$\mu\nabla\left(g\frac{\nabla u}{\left|\nabla u\right|}\right)+f=0,\quad\text{in }\Omega,\qquad\frac{\partial u}{\partial\vec{n}}=0,\quad\text{on }\partial\Omega \tag{5.6}$$

where $f=-r-\theta P_d-\alpha\nu'\left(u\right)$. When u is fixed, the intensity values c_1,c_2 are updated by

$$c_1(u)=\frac{\int_\Omega uz\ d\Omega}{\int_\Omega u\ d\Omega},\qquad c_2(u)=\frac{\int_\Omega\left(1-u\right)z\ d\Omega}{\int_\Omega\left(1-u\right)\ d\Omega}.$$

Notice that the nonlinear coefficient of equation (5.6) may have a zero denominator

where the equation is not defined. A commonly adopted idea to deal with this is to introduce a positive parameter β to (5.6), so the new Euler Lagrange equation becomes

$$\mu\nabla\left(g\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}\right)+f=0,\quad in\quad \Omega;\qquad \frac{\partial u}{\partial\vec{n}}=0,\quad on\quad \partial\Omega$$

which corresponds to minimise the following differentiable form of (5.5)

$$\min_u CDSS\left(u,c_1,c_2\right)=\mu\int_\Omega g\sqrt{|\nabla u|^2+\beta}\,d\Omega+\int_\Omega ru\,d\Omega+\theta\int_\Omega P_d u\,d\Omega+\alpha\int_\Omega \nu\left(u\right)d\Omega. \tag{5.7}$$

According to [129, 128], the standard AOS which generally assumes f is not dependent on u is not adequate to solve the model. This mainly because the term $\nu'\left(u\right)$ in f does depend on u, which can lead to stability restriction on time step size t. Moreover, the shape of $\nu'\left(u\right)$ means that changes in f between iterations are problematic near $u=0$ and $u=1$, as small changes in u produce large changes in f. In order to tackle the problem, they proposed a modified version of AOS algorithm to solve the model by taking the approximation of $\nu'\left(u\right)$ which based on its linear part.

A successful segmentation result can be obtained depending on suitable combination of parameter μ, θ and the set of marker points defined by a user. For a simple image such as synthetic images, this task of parameters selection is easy and one can get a good segmentation result. However, for real life images, it is non-trivial to determine a suitable combination of parameters μ and θ. It becomes more challenging if a model is sensitive to μ and θ where only a small range of the values work to give high segmentation quality. Hence, a more robust model that is less dependent on the parameters needs to be developed. In addition, to process images of large size, fast iterative solvers need to be developed as well. This paper is motivated by these two problems.

We refer to the CDSS model solved using the modified AOS as **SC0**.

5.3 A reformulated CDSS model

We now present our work on a reformulation of the CDSS model in the primal-dual framework which allows us to "ignore" the penalty function $\nu\left(u\right)$, otherwise creating problems of parameter sensitivity. We remark that similar use of the primal-dual idea can be found in [40] and [99]. To see more background of this framework, refer to the convex regularisation approach by Bresson et al. [20], Chambolle [26], and others [8, 9, 35, 24].

Our starting point is to rewrite (5.5) as follows:

$$\begin{aligned}\min_{u,w} J\left(u,w\right)= \quad & \mu\int_\Omega |\nabla u|_g d\Omega+\int_\Omega rw\,d\Omega+\theta\int_\Omega P_d w\,d\Omega\\ & +\alpha\int_\Omega \nu\left(w\right)\,d\Omega+\frac{1}{2\rho}\int_\Omega\left(u-w\right)^2 d\Omega\end{aligned} \tag{5.8}$$

where w is the new and dual variable, the right-most term enforces $w \approx u$ for sufficiently small $\rho > 0$ and $|\nabla u|_g = g(|\nabla z|)|\nabla u|$. One can observe that if $w = u$, the dual formulation is reduced to the original CDSS model [129].

After introducing the term $(u-w)^2$, it is important to note that convexity still holds with respect to u and w (otherwise finding the global minimum cannot be guaranteed). This can be shown below. Write the functional (5.8) as the sum of two terms:

$$J(u,w) = S(u,w) + Q(u,w),\ S(u,w) = \int_\Omega \frac{1}{2\rho}(u-w)^2 d\Omega,\ TV_g(u) = \int_\Omega |\nabla u|_g\, d\Omega$$
$$Q(u,w) = TV_g(u) + \int_\Omega (r + \theta P_d)\, w d\Omega + \alpha \int_\Omega \nu(w)\, d\Omega.$$

For the functional $Q(u,w)$, we can show that the weighted total variation term $TV_g(u)$ is convex below. The remaining two terms (depending on w only) are known to be convex from [129, 128]. By definition of convex functions, showing that the weighted total variation is a convex can be done directly. Let $u_1 \neq u_2$ be two functions and $\varphi \in [0,1]$. Then

$$\begin{aligned} TV_g(\varphi u_1 + (1-\varphi)u_2) &= \int_\Omega |\nabla(\varphi u_1 + (1-\varphi)u_2)|_g\, d\Omega \\ &= \int_\Omega |\varphi \nabla u_1 + (1-\varphi)\nabla u_2|_g\, d\Omega \\ &\leq \varphi \int_\Omega |\nabla u_1|_g\, d\Omega + (1-\varphi)\int_\Omega |\nabla u_2|_g\, d\Omega \\ &= \varphi TV_g(u_1) + (1-\varphi)TV_g(u_2). \end{aligned}$$

Similarly, for the functional $S(u,w)$, let $u, w : \Omega \subseteq \mathbb{R}^2 \to \mathbb{R}$ and $u_1 \neq u_2 \neq u_3 \neq u_4$. Then

$$\begin{aligned} S[\varphi(u_1,u_2) + (1-\varphi)(u_3,u_4)] &= S[\varphi u_1 + (1-\varphi)u_3, \varphi u_2 + (1-\varphi)u_4] \\ &= \int_\Omega [\varphi u_1 + (1-\varphi)u_3 - \varphi u_2 - (1-\varphi)u_4]^2 d\Omega \\ &= \int_\Omega [\varphi(u_1-u_2) + (1-\phi)(u_3-u_4)]^2 d\Omega \\ &\leq \varphi \int_\Omega (u_1-u_2)^2 d\Omega + (1-\varphi)\int_\Omega (u_3-u_4)^2 d\Omega \\ &= \varphi S(u_1,u_2) + (1-\varphi)S(u_3,u_4). \end{aligned}$$

Alternatively, the $Hessian\left[(u-w)^2\right] = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$. Clearly the principal minors are $\Delta_1 = 2, \quad \Delta_2 = 0$ which indicates that the $Hessian[(u-w)^2]$ is positive semidefinite and so $S(u,w)$ is convex.

As the sum of two convex functions Q, S is also convex, thus $J(u,w)$ is convex.

Using the property that J is differentiable, consequently, the unique minimiser can be computed by minimising J with respect to u and w separately, iterating the process until convergence [20, 26]. Thus, the following minimisation problems are considered:

i). when w is given: $\min_u\ J_1(u,w) = \mu \int_\Omega |\nabla u|_g d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2 d\Omega$;

ii). when u is given: $\min_{w} \; J_2(u,w) = \int_\Omega r w d\Omega + \theta \int_\Omega P_d w d\Omega + \alpha \int_\Omega \nu(w)\, d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2\, d\Omega$.

Next consider how to simplify J_2 further and drop its α term. To this end, we make use of the following proposition:

Proposition 5.3.1. *The solution of* $\min_w J_2$ *is given by:*

$$w = \min\left\{\max\left\{u(x) - \rho r - \rho\theta P_d, 0\right\}, 1\right\}. \tag{5.9}$$

Proof: Assume that α has been chosen large enough compared to $\|f\|_{L^\infty}$ so that the exact penalty formulation holds. We now consider the w-minimisation of the form $\min_w \int_\Omega \left(\alpha\nu(w) + \frac{1}{2\rho}(u-w)^2 + wF(x)\right) d\Omega$, where the function F is independent of w. We use the claim made by [20].

Claim [20]**:** If $u(x) \in [0,1]$ for all x, then so is $w(x)$ after the w-minimisation. Conversely, if $w(x) \in [0,1]$ for all x, then so is $u(x)$ after the u-minimisation.

This claim allows us to "ignore" the $\nu(w)$ terms: on one hand, its presence in the energy is equivalent to cutting off $w(x)$ at 0 and 1. On the other hand, if $w(x) \in [0,1]$, then the above w-minimisation can be written in this equivalence form: $\min_{w\in[0,1]} \int_\Omega \left(\frac{1}{2\rho}(u-w)^2 + wF(x)\right) d\Omega$. Consequently, the point-wise optimal $w(x)$ is found as $\frac{1}{\rho}(u-w) = F(x) \Rightarrow w = u - \rho F(x)$. Thus the w-minimisation can be achieved through the following update:
$w = \min\left\{\max\left\{u(x) - \rho F(x), 0\right\}, 1\right\}$. For $\min_w J_2$, let $F(x) = r + \theta P_d$. Hence, we deduce the result for w. □

Therefore, our new model is defined as

$$\min_{u,w\in[0,1]} J(u,w) = \mu\int_\Omega |\nabla u|_g d\Omega + \int_\Omega rw\, d\Omega + \theta\int_\Omega P_d w\, d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2\, d\Omega.$$

In alternating minimisation form, the new formulation is equivalent to solve the following

$$\min_{u} \; J_1(u,w) = \mu\int_\Omega |\nabla u|_g d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2 d\Omega, \tag{5.10}$$

$$\min_{w\in[0,1]} \; J_2(u,w) = \int_\Omega rw\, d\Omega + \theta\int_\Omega P_d w\, d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2\, d\Omega. \tag{5.11}$$

Notice that the term $\nu(w)$ is dropped in (5.11) and the explicit solution is given in (5.9) that is hopefully the new resulting model becomes less sensitive to parameter's choice. Now it only remains to discuss how to solve (5.10).

5.4 An optimisation based multilevel algorithm

This section presents our multilevel formulation for two convex models: first the CDSS model (5.7) (for later use in comparisons) and then our newly proposed primal-dual

model in (5.10)-(5.11).

For simplicity, we shall assume $n = 2^L$ for a given image z of size $n \times n$. The standard coarsening defines $L+1$ levels: $k = 1$ (finest) $, 2, ..., L,\ L+1$ (coarsest) such that level k has $\tau_k \times \tau_k$ "superpixels" with each "superpixels" having pixels $b_k \times b_k$ where $\tau_k = n/2^{k-1}$ and $b_k = 2^{k-1}$. See Figure 4.2(a-e) of Chapter 4 for the details illustration.

5.4.1 A multilevel algorithm for CDSS

Our goal is to solve (5.7) using a multilevel method in discretise-optimise scheme without approximation of $\nu'(u)$. The finite difference method is used to discretise (5.7) as done in related works [24, 33]. The discretised version of (5.7) is given by

$$\begin{aligned}\min_u CDSS\,(u, c_1, c_2) &\equiv \min_u CDSS^a\,(u_{1,1}, u_{2,1}, ..., u_{i-1,j}, u_{i,j}, u_{i+1,j}, ..., u_{n,n}, c_1, c_2) \\ &= \bar{\mu} \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} g_{i,j}\sqrt{(u_{i,j} - u_{i,j+1})^2 + (u_{i,j} - u_{i+1,j})^2 + \beta} \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{n}\Big((c_1 - z_{i,j})^2 - (c_2 - z_{i,j})^2\Big)u_{i,j} + \theta\sum_{i=1}^{n}\sum_{j=1}^{n} P_{d_{i,j}}u_{i,j} + \alpha\sum_{i=1}^{n}\sum_{j=1}^{n}\nu_{i,j}\end{aligned} \tag{5.12}$$

where $\bar{\mu} = \frac{\mu}{h}$, $c_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} z_{i,j}u_{i,j} \big/ \sum_{i=1}^{n}\sum_{j=1}^{n} u_{i,j}$, $c_2 = \sum_{i=1}^{n}\sum_{j=1}^{n} z_{i,j}\,(1 - u_{i,j}) \big/ \sum_{i=1}^{n}\sum_{j=1}^{n}(1 - u_{i,j})$,

$$\nu_{i,j} = \left[\sqrt{(2u_{i,j} - 1)^2 + \varepsilon} - 1\right]\Big(\frac{1}{2} + \frac{1}{\pi}\arctan\frac{\sqrt{(2u_{i,j} - 1)^2 + \varepsilon} - 1}{\varepsilon}\Big),$$

$$g_{i,j} = g(x_i, y_j) \quad \text{and} \quad P_{d_{i,j}} = (x_i, y_j).$$

Here u denotes a row vector.

As a prelude to multilevel methods, minimise (5.12) by a coordinate descent method (also known as relaxation algorithm) on the finest level 1:

$$\begin{cases} \text{Given} & u^{(0)} = \left(u_{i,j}^{(0)}\right) \ \text{ with } m = 0, \\ \text{Solve} & u_{i,j}^{(m+1)} = \underset{u_{i,j}\in\mathbb{R}}{\arg\min}\, CDSS^{loc}\,(u_{i,j}, c_1, c_2) \ \text{ for } i, j = 1, 2, ...n, \\ & \text{Repeat the above steps with } m = m + 1 \text{ until stopped.} \end{cases} \tag{5.13}$$

Here equation (5.13) is simply obtained by expanding and simplifying the main model

in (5.12) i.e.

$$
\begin{aligned}
&CDSS^{loc}\left(u_{i,j},c_1,c_2\right)\\
&\equiv CDSS^{a}\left(u_{1,1}^{(m)},u_{2,1}^{(m)},...,u_{i-1,j}^{(m)},u_{i,j},u_{i+1,j}^{(m)},...,u_{m,n}^{(m)},c_1,c_2\right)-CDSS^{(m)}\\
&\quad=\bar{\mu}\Bigg[g_{i,j}\sqrt{\left(u_{i,j}-u_{i+1,j}^{(m)}\right)^2+\left(u_{i,j}-u_{i,j+1}^{(m)}\right)^2+\beta}\\
&\quad+g_{i-1,j}\sqrt{\left(u_{i,j}-u_{i-1,j}^{(m)}\right)^2+\left(u_{i-1,j}^{(m)}-u_{i-1,j+1}^{(m)}\right)^2+\beta}\\
&\quad+g_{i,j-1}\sqrt{\left(u_{i,j}-u_{i,j-1}^{(m)}\right)^2+\left(u_{i,j-1}^{(m)}-u_{i+1,j-1}^{(m)}\right)^2+\beta}\Bigg]\\
&\quad+u_{i,j}\left((c_1-z_{i,j})^2-(c_2-z_{i,j})^2\right)+\theta P_{d_{i,j}}u_{i,j}+\alpha\left(\nu_{i,j}\right)
\end{aligned}
$$

with Neumann's boundary condition applied where $CDSS^{(m)}$ denotes the sum of all terms in $CDSS^{a}$ that do not involve $u_{i,j}$. Clearly one seems that this is a coordinate descent method. It should be remarked that the formulation in (5.13) is based on the work in [24] and [33].

The Newton method is used to solve the one-dimensional problem from (5.13) by iterating $u^{(m)}\rightarrow u\rightarrow u^{(m+1)}$:

$$
\begin{aligned}
&\bar{\mu}g_{i,j}\frac{2u_{i,j}-u_{i+1,j}^{(m)}-u_{i,j+1}^{(m)}}{\sqrt{\left(u_{i,j}-u_{i+1,j}^{(m)}\right)^2+\left(u_{i,j}-u_{i,j+1}^{(m)}\right)^2+\beta}}+\bar{\mu}g_{i-1,j}\frac{u_{i,j}-u_{i-1,j}^{(m)}}{\sqrt{\left(u_{i,j}-u_{i-1,j}^{(m)}\right)^2+\left(u_{i-1,j}^{(m)}-u_{i-1,j+1}^{(m)}\right)^2+\beta}}\\
&+\bar{\mu}g_{i,j-1}\frac{u_{i,j}-u_{i,j-1}^{(m)}}{\sqrt{\left(u_{i,j}-u_{i,j-1}^{(m)}\right)^2+\left(u_{i,j-1}^{(m)}-u_{i+1,j-1}^{(m)}\right)^2+\beta}}+\left((c_1-z_{i,j})^2-(c_2-z_{i,j})^2\right)\\
&+\theta P_{d_{i,j}}+\alpha{\nu_{i,j}}'=0
\end{aligned}
$$

giving rise to the form

$$
u_{i,j}^{new}=u_{i,j}^{old}-T^{old}/B^{old} \tag{5.14}
$$

where

$$
\begin{aligned}
T^{old}=&\frac{(\bar{\mu}g_{i,j})(2u_{i,j}^{old}-u_{i+1,j}^{(m)}-u_{i,j+1}^{(m)})}{\sqrt{\left(u_{i,j}^{old}-u_{i+1,j}^{(m)}\right)^2+\left(u_{i,j}^{old}-u_{i,j+1}^{(m)}\right)^2+\beta}}+\frac{(\bar{\mu}g_{i-1,j})(u_{i,j}^{old}-u_{i-1,j}^{(m)})}{\sqrt{\left(u_{i,j}^{old}-u_{i-1,j}^{(m)}\right)^2+\left(u_{i-1,j}^{(m)}-u_{i-1,j+1}^{(m)}\right)^2+\beta}}\\
&+\frac{(\bar{\mu}g_{i,j-1})(u_{i,j}^{old}-u_{i,j-1}^{(m)})}{\sqrt{\left(u_{i,j}^{old}-u_{i,j-1}^{(m)}\right)^2+\left(u_{i,j-1}^{(m)}-u_{i+1,j-1}^{(m)}\right)^2+\beta}}+\left((c_1-z_{i,j})^2-(c_2-z_{i,j})^2\right)\\
&+\theta P_{d_{i,j}}+\alpha{\nu_{i,j}}'^{(old)}
\end{aligned}
$$

$$
\begin{aligned}
B^{old}=&\frac{2\bar{\mu}g_{i,j}}{\sqrt{\left(u_{i,j}^{old}-u_{i+1,j}^{(m)}\right)^2+\left(u_{i,j}^{old}-u_{i,j+1}^{(m)}\right)^2+\beta}}-\frac{\bar{\mu}g_{i,j}\left(2u_{i,j}^{old}-u_{i+1,j}^{(m)}-u_{i,j+1}^{(m)}\right)^2}{\sqrt{\left(\left(u_{i,j}^{old}-u_{i+1,j}^{(m)}\right)^2+\left(u_{i,j}^{old}-u_{i,j+1}^{(m)}\right)^2+\beta\right)^{\frac{3}{2}}}}\\
&+\frac{\bar{\mu}g_{i-1,j}}{\sqrt{\left(u_{i,j}^{old}-u_{i-1,j}^{(m)}\right)^2+\left(u_{i-1,j}^{(m)}-u_{i-1,j+1}^{(m)}\right)^2+\beta}}-\frac{\bar{\mu}g_{i-1,j}\left(u_{i,j}^{old}-u_{i-1,j}^{(m)}\right)^2}{\sqrt{\left(\left(u_{i,j}^{old}-u_{i-1,j}^{(m)}\right)^2+\left(u_{i-1,j}^{(m)}-u_{i-1,j+1}^{(m)}\right)^2+\beta\right)^{\frac{3}{2}}}}\\
&+\frac{\bar{\mu}g_{i,j-1}}{\sqrt{\left(u_{i,j}^{old}-u_{i,j-1}^{(m)}\right)^2+\left(u_{i,j-1}^{(m)}-u_{i+1,j-1}^{(m)}\right)^2+\beta}}-\frac{\bar{\mu}g_{i,j-1}\left(u_{i,j}^{old}-u_{i,j-1}^{(m)}\right)^2}{\sqrt{\left(\left(u_{i,j}^{old}-u_{i,j-1}^{(m)}\right)^2+\left(u_{i,j-1}^{(m)}-u_{i+1,j-1}^{(m)}\right)^2+\beta\right)^{\frac{3}{2}}}}\\
&+\alpha{\nu_{i,j}}''^{(old)}.
\end{aligned}
$$

To develop a multilevel method for this coordinate descent method, we interpret solving (5.13) as looking for the best correction constant $\hat{c}$ at the current approximation $u_{i,j}^{(m)}$ on level 1 (the finest level) that minimises for c i.e.

$$\min_{u_{i,j}\in\mathbb{R}} CDSS^{loc}\left(u_{i,j}, c_1, c_2\right) = \min_{c\in\mathbb{R}} CDSS^{loc}\left(u_{i,j}^{(m)} + c, c_1, c_2\right).$$

Hence, we may rewrite (5.13) in an equivalent form:

$$\begin{cases} \text{Given} & u^{(0)} = \left(u_{i,j}^{(0)}\right) \quad \text{with } m = 0, \\ \text{Solve} & \hat{c} = \arg\min_{c\in\mathbb{R}} CDSS^{loc}\left(u_{i,j}^{(m)} + c, c_1, c_2\right) \text{ for } i, j = 1, 2, ...n, \\ \text{Update} & u_{i,j}^{(m+1)} = u_{i,j}^{(m)} + \hat{c}, \\ & \text{Repeat the above steps with } m = m + 1 \text{ until stopped.} \end{cases} \tag{5.15}$$

Let's set the following: $b = 2^{k-1}$, $k_1 = (i-1)\,b + 1$, $k_2 = ib$, $\ell_1 = (j-1)\,b + 1$, $\ell_2 = jb$, and $c = (c_{i,j})$. Denoted the current $\tilde{u}$ then, the computational stencil involving c on level k can be shown as follows

$$\begin{array}{c|c|c|c|c}
\vdots & \vdots & \cdots & \vdots & \vdots \\
\widetilde{u}_{k_1-1,\ell_2+1} + c_{i-1,j+1} & \widetilde{u}_{k_1,\ell_2+1} + c_{i,j+1} & \cdots & \widetilde{u}_{k_2,\ell_2+1} + c_{i,j+1} & \widetilde{u}_{k_2+1,\ell_2+1} + c_{i+1,j+1} \\
\hline
\widetilde{u}_{k_1-1,\ell_2} + c_{i-1,j} & \widetilde{u}_{k_1,\ell_2} + c_{i,j} & \cdots & \widetilde{u}_{k_2,\ell_2} + c_{i,j} & \widetilde{u}_{k_2+1,\ell_2} + c_{i+1,j} \\
\cdots & \vdots & \cdots & \vdots & \cdots \\
\widetilde{u}_{k_1-1,\ell_1} + c_{i-1,j} & \widetilde{u}_{k_1,\ell_1} + c_{i,j} & \cdots & \widetilde{u}_{k_2,\ell_1} + c_{i,j} & \widetilde{u}_{k_2+1,\ell_1} + c_{i+1,j} \\
\hline
\widetilde{u}_{k_1-1,\ell_1-1} + c_{i-1,j-1} & \widetilde{u}_{k_1,\ell_1-1} + c_{i,j-1} & \cdots & \widetilde{u}_{k_2,\ell_1-1} + c_{i,j-1} & \widetilde{u}_{k_2+1,\ell_1-1} + c_{i+1,j-1} \\
\vdots & \vdots & \cdots & \vdots & \vdots
\end{array} \tag{5.16}$$

The illustration shown above is consistent with Figure 4.2 (f) and the key point is that interior pixels do not involve $c_{i,j}$ in the formulation's first nonlinear term. This is because the finite differences are not changed at interior pixels by the same update as in

$$\begin{aligned} &\sqrt{\left(\tilde{u}_{k,l} + c_{i,j} - \tilde{u}_{k+1,l} - c_{i,j}\right)^2 + \left(\tilde{u}_{k,l} + c_{i,j} - \tilde{u}_{k,l+1} - c_{i,j}\right)^2 + \beta} \\ &\qquad = \sqrt{\left(\tilde{u}_{k,l} - \tilde{u}_{k+1,l}\right)^2 + \left(\tilde{u}_{k,l} - \tilde{u}_{k,l+1}\right)^2 + \beta}. \end{aligned}$$

Then, minimising for c, the problem (5.15) is equivalent to minimise the following

$$
\begin{aligned}
F_{SC1}\left(c_{i,j}\right) = & \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_1-1,\ell} - \tilde{u}_{k_1,\ell}\right)\right]^2 + \left(\tilde{u}_{k_1-1,\ell} - \tilde{u}_{k_1-1,\ell+1}\right)^2 + \beta} \\
& + \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k,\ell_2+1} - \tilde{u}_{k,\ell_2}\right)\right]^2 + \left(\tilde{u}_{k,\ell_2} - \tilde{u}_{k+1,\ell_2}\right)^2 + \beta} \\
& + \bar{\mu} g_{k_2,\ell_2} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_2,\ell_2+1} - \tilde{u}_{k_2,\ell_2}\right)\right]^2 + \left[c_{i,j} - \left(\tilde{u}_{k_2+1,\ell_2} - \tilde{u}_{k_2,\ell_2}\right)\right]^2 + \beta} \\
& + \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_2+1,\ell} - \tilde{u}_{k_2,\ell}\right)\right]^2 + \left(\tilde{u}_{k_2,\ell} - \tilde{u}_{k_2,\ell+1}\right)^2 + \beta} \\
& + \bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k,\ell_1-1} - \tilde{u}_{k,\ell_1}\right)\right]^2 + \left(\tilde{u}_{k,\ell_1-1} - \tilde{u}_{k+1,\ell_1-1}\right)^2 + \beta} \\
& + \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(\tilde{u}_{k,\ell} + c_{i,j}\right) \left(\left(c_1 - z_{k,\ell}\right)^2 - \left(c_2 - z_{k,\ell}\right)^2\right) \\
& + \theta \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(\tilde{u}_{k,\ell} + c_{i,j}\right) P_{d_{k,\ell}} + \alpha \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \nu\left(\tilde{u}_{k,\ell} + c_{i,j}\right)
\end{aligned}
\tag{5.17}
$$

where the third term may be simplified using $(c-a)^2 + (c-b)^2 + \beta = 2\left(c - \frac{a+b}{2}\right)^2 + 2\left(\frac{a-b}{2}\right)^2 + \beta$. Further the local minimisation problem for block (i,j) on level k with respect to $c_{i,j}$ amounts to minimising the following equivalent functional

$$
\begin{aligned}
F_{SC1}\left(c_{i,j}\right) = & \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{\left(c_{i,j} - h_{k_1-1,\ell}\right)^2 + \upsilon_{k_1-1,\ell}^2 + \beta} \\
& + \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_2}\right)^2 + h_{k,\ell_2}^2 + \beta} \\
& + \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{\left(c_{i,j} - h_{k_2,\ell}\right)^2 + \upsilon_{k_2,\ell}^2 + \beta} \\
& + \bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_1-1}\right)^2 + h_{k,\ell_1-1}^2 + \beta} \\
& + \bar{\mu} \sqrt{2} g_{k_2,\ell_2} \sqrt{\left(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2}\right)^2 + \bar{h}_{k_2,\ell_2}^2 + \frac{\beta}{2}} \\
& + \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(c_{i,j}\right) \left(\left(c_1 - z_{k,\ell}\right)^2 - \left(c_2 - z_{k,\ell}\right)^2\right) \\
& + \theta \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(\tilde{u}_{k,\ell} + c_{i,j}\right) P_{d_{k,\ell}} + \alpha \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \nu\left(\tilde{u}_{k,\ell} + c_{i,j}\right)
\end{aligned}
\tag{5.18}
$$

where we have used the following notation (which will be used later also):

$$\begin{array}{lll}
h_{k,\ell} = \tilde{u}_{k+1,\ell} - \tilde{u}_{k,\ell}, & v_{k,\ell} = \tilde{u}_{k,\ell+1} - \tilde{u}_{k,\ell}, & v_{k_2,\ell_2} = \tilde{u}_{k_2,\ell_2+1} - \tilde{u}_{k_2,\ell_2}, \\
h_{k_2,\ell_2} = \tilde{u}_{k_2+1,\ell_2} - \tilde{u}_{k_2,\ell_2}, & \bar{v}_{k_2,\ell_2} = \frac{v_{k_2,\ell_2}+h_{k_2,\ell_2}}{2}, & \bar{h}_{k_2,\ell_2} = \frac{v_{k_2,\ell_2}-h_{k_2,\ell_2}}{2}, \\
h_{k_1-1,\ell} = \tilde{u}_{k_1,\ell} - \tilde{u}_{k_1-1,\ell}, & v_{k_1-1,\ell} = \tilde{u}_{k_1-1,\ell+1} - \tilde{u}_{k_1-1,\ell}, & v_{k,\ell_2} = \tilde{u}_{k,\ell_2+1} - \tilde{u}_{k,\ell_2}, \\
h_{k,\ell_2} = \tilde{u}_{k+1,\ell_2} - \tilde{u}_{k,\ell_2}, & h_{k_2,\ell} = \tilde{u}_{k_2+1,\ell} - \tilde{u}_{k_2,\ell}, & v_{k_2,\ell} = \tilde{u}_{k_2,\ell+1} - \tilde{u}_{k_2,\ell}, \\
v_{k,\ell_1-1} = \tilde{u}_{k,\ell_1} - \tilde{u}_{k,\ell_1-1}, & h_{k,\ell_1-1} = \tilde{u}_{k+1,\ell_1-1} - \tilde{u}_{k,\ell_1-1}. &
\end{array} \tag{5.19}$$

For solution on the coarsest level, we look for a **single** constant update for the current approximation $\tilde{u}$ that is

$$\begin{aligned}
\min_c \{F_{SC1}(\tilde{u}+c) = & \sum_{i=1}^{n}\sum_{j=1}^{n}(\tilde{u}_{i,j}+c)\left((c_1-z_{i,j})^2-(c_2-z_{i,j})^2\right) \\
& +\bar{\mu}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} g_{i,j}\sqrt{(\tilde{u}_{i,j}+c-\tilde{u}_{i,j+1}-c)^2+(\tilde{u}_{i,j}+c-\tilde{u}_{i+1,j}-c)^2+\beta} \\
& +\theta\sum_{i=1}^{n}\sum_{j=1}^{n} P_{d_{i,j}}(\tilde{u}_{i,j}+c)+\alpha\sum_{i=1}^{n}\sum_{j=1}^{n}\nu(\tilde{u}_{i,j}+c)\}
\end{aligned}$$

which is equivalent to

$$\begin{aligned}
\min_c \{\, F_{SC1}(\tilde{u}+c) = & \sum_{i=1}^{n}\sum_{j=1}^{n}(\tilde{u}_{i,j}+c)\left((c_1-z_{i,j})^2-(c_2-z_{i,j})^2\right) \\
& +\theta\sum_{i=1}^{n}\sum_{j=1}^{n} P_{d_{i,j}}(\tilde{u}_{i,j}+c)+\alpha\sum_{i=1}^{n}\sum_{j=1}^{n}\nu(\tilde{u}_{i,j}+c)\}.
\end{aligned} \tag{5.20}$$

The solutions of the above local minimisation problems, solved using a Newton method as in (5.14) or a fixed point method for t iterations (inner iteration), defines the update solution $u = u + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_k \times b_k$ block on level k as illustrated in (5.16). Then we obtain a multilevel method if we cycle through all levels and all blocks on each level until the relative error in two consecutive cycles (outer iteration) is smaller than *tol* or the maximum number of cycle, *maxit* is reached.

Finally our proposed multilevel method for CDSS is summarised in Algorithm 5. We will use the term **SC1** to refer this multilevel Algorithm 5.

In order to get fast convergence, it is recommended to start updating our multilevel algorithm from the fine level to the coarse level. In a separate experiment we found that if we adjust the coarse structure before the fine level, the convergence is slower. In addition, we recommend the value of inner iteration $t = 1$ is used to update the algorithm in a fast manner.

Algorithm 5 SC1 – Multilevel algorithm for the CDSS model

Given z, an initial guess u, the stop tolerance (tol), and maximum multilevel cycle (maxit) with $L+1$ levels,

1) *Set $\tilde{u} = u$.*

2) *Smooth for t iteration the approximation on the finest level 1 that is solve* (5.13) *for $i, j = 1, 2, ...n$*

3) *Iterate for t times on each coarse level $k = 2, 3, ...L, L+1$:*
   > If $k \leq L$, compute the minimiser c of (5.18)
   > Solve (5.20) *on the coarsest level $k = L+1$*
   > Add the correction $u = u + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_k \times b_k$ block on level k as illustrated in (5.16).

4) *Check for convergence using the above criteria. If not satisfied, return to Step 1. Otherwise exit with solution $u = \tilde{u}$.*

5.4.2 A multilevel algorithm for the proposed model

We now consider our main model as expressed by (5.10)–(5.11). Minimisations of J is with respect to u in (5.10) and w in (5.11) respectively. The solution of (5.11) can be obtained analytically following Proposition 5.3.1. It remains to develop a multilevel algorithm to solve (5.10).

Similar to the last subsection, the discretised form of the functional $J_1(u, w)$ of problem (5.10) is as follows:

$$\min_u J_1(u, w), \tag{5.21}$$

where

$$J_1(u, w) = \bar{\mu} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} g_{i,j} \sqrt{(u_{i,j} - u_{i,j+1})^2 + (u_{i,j} - u_{i+1,j})^2 + \beta} + \frac{1}{2\rho} \sum_{i=1}^{n} \sum_{j=1}^{n} (u_{i,j} - w_{i,j})^2 \tag{5.22}$$

Clearly this is a much simpler functional than the CDSS model (5.12) so the method can be similarly developed.

Consider the minimisation of (5.22) by the coordinate descent method on the finest level 1:

$$\begin{cases} \text{Given} & u^{(0)} = \left(u_{i,j}^{(0)}\right) \text{ with } m = 0, \\ \text{Solve} & u_{i,j}^{(m+1)} = \underset{u_{i,j} \in \mathbb{R}}{\arg\min} J_1^{loc}(u_{i,j}, c_1, c_2) \text{ for } i, j = 1, 2, ...n, \\ & \text{Repeat the above steps with } m = m + 1 \text{ until stopped.} \end{cases} \tag{5.23}$$

where,

$$J_1^{loc}(u_{i,j}, c_1, c_2) = J_1 - J_0 = \begin{aligned} &\bar{\mu} g_{i,j} \sqrt{\left(u_{i,j} - u_{i+1,j}^{(m)}\right)^2 + \left(u_{i,j} - u_{i,j+1}^{(m)}\right)^2 + \beta} \\ &+\bar{\mu} g_{i-1,j} \sqrt{\left(u_{i,j} - u_{i-1,j}^{(m)}\right)^2 + \left(u_{i-1,j}^{(m)} - u_{i-1,j+1}^{(m)}\right)^2 + \beta} \\ &+\bar{\mu} g_{i,j-1} \sqrt{\left(u_{i,j} - u_{i,j-1}^{(m)}\right)^2 + \left(u_{i,j-1}^{(m)} - u_{i+1,j-1}^{(m)}\right)^2 + \beta} \\ &+ \tfrac{1}{2\rho}(u_{i,j} - w_{i,j})^2. \end{aligned}$$

The term J_0 refers to a collection of all terms that are not dependent on $u_{i,j}$. For $u_{i,j}$ at the boundary, Neumann's condition is used. Note that each subproblem in (5.23) is only one dimensional, which is the key to the efficiency of our new method.

To introduce the multilevel algorithm, it is of interest to rewrite (5.23) in an equivalent form:

$$\hat{c} = \arg\min_{c \in \mathbb{R}} J_1^{loc}\left(u_{i,j}^{(m)} + c, c_1, c_2\right), \quad u_{i,j}^{(m+1)} = u_{i,j}^{(m)} + \hat{c} \text{ for } i, j = 1, 2, ..., n. \tag{5.24}$$

Using the stencil in (5.16), the problem (5.24) is equivalent to minimise the following

$$\begin{aligned} F_2(c_{i,j}) = &\bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1,\ell} \sqrt{[c_{i,j} - (\tilde{u}_{k_1-1,\ell} - \tilde{u}_{k_1,\ell})]^2 + (\tilde{u}_{k_1-1,\ell} - \tilde{u}_{k_1-1,\ell+1})^2 + \beta} \\ &+\bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{[c_{i,j} - (\tilde{u}_{k,\ell_2+1} - \tilde{u}_{k,\ell_2})]^2 + (\tilde{u}_{k,\ell_2} - \tilde{u}_{k+1,\ell_2})^2 + \beta} \\ &+\bar{\mu} g_{k_2,\ell_2} \sqrt{[c_{i,j} - (\tilde{u}_{k_2,\ell_2+1} - \tilde{u}_{k_2,\ell_2})]^2 + [c_{i,j} - (\tilde{u}_{k_2+1,\ell_2} - \tilde{u}_{k_2,\ell_2})]^2 + \beta} \\ &+\bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{[c_{i,j} - (\tilde{u}_{k_2+1,\ell} - \tilde{u}_{k_2,\ell})]^2 + (\tilde{u}_{k_2,\ell} - \tilde{u}_{k_2,\ell+1})^2 + \beta} \\ &+\bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{[c_{i,j} - (\tilde{u}_{k,\ell_1-1} - \tilde{u}_{k,\ell_1})]^2 + (\tilde{u}_{k,\ell_1-1} - \tilde{u}_{k+1,\ell_1-1})^2 + \beta} \\ &+\frac{1}{2\rho} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} (u_{k,\ell} + c_{i,j} - w_{k,\ell})^2. \end{aligned} \tag{5.25}$$

After some algebraic manipulation to simplify (5.25), we arrive at the following

$$\begin{aligned} F_2\left(c_{i,j}\right) = & \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{\left(c_{i,j} - h_{k_1-1,\ell}\right)^2 + \upsilon_{k_1-1,\ell}^2 + \beta} \\ & + \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_2}\right)^2 + h_{k,\ell_2}^2 + \beta} \\ & + \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{\left(c_{i,j} - h_{k_2,\ell}\right)^2 + \upsilon_{k_2,\ell}^2 + \beta} \\ & + \bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_1-1}\right)^2 + h_{k,\ell_1-1}^2 + \beta} \\ & + \bar{\mu} \sqrt{2} g_{k_2,\ell_2} \sqrt{\left(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2}\right)^2 + \bar{h}_{k_2,\ell_2}^2 + \frac{\beta}{2}} \\ & + \frac{1}{2\rho} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(u_{k,\ell} + c_{i,j} - w_{k,\ell}\right)^2. \end{aligned} \tag{5.26}$$

On the coarsest level $(L+1)$, a **single** constant update for the current $\tilde{u}$ is given as

$$\min_c \left\{ F_2\left(\tilde{u} + c\right) = \frac{1}{2\rho} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(u_{i,j} + c - w_{i,j}\right)^2 \right\} \tag{5.27}$$

which has a simple and explicit solution.

Then, we obtain a multilevel method if we cycle through all levels and all blocks on each level. The process is stopped if the relative error in two consecutive cycles (outer iteration) is smaller than *tol* or the maximum number of cycle, *maxit* is reached.

The overall procedure to solve the new primal-dual model is given in Algorithm 6. We will use the term **SC2** to refer this algorithm to solve the proposed model expressed in (5.10) and (5.11).

Again, in order to update the algorithm in a fast manner, we recommend to adjust the fine level before the coarse level and to use the inner iteration $t = 1$.

5.5 A new variant of the multilevel algorithm SC2

Our above proposed method defines a sequence of search directions based in a multilevel setting for an optimisation problem. We now modify it so that the new algorithm has a formal decaying property.

Denote the functional in (5.22) by $g(u) : \mathbb{R}^{n^2} \to \mathbb{R}$ and represent each subproblem by

$$c^* = \underset{c \in \mathbb{R}}{\operatorname{argmin}}\, g(u^\ell + c p^\ell), \quad u^{\ell+1} = u^\ell + c^* p^\ell, \quad p^\ell = \tilde{\mathbf{e}}^{\ell (\operatorname{mod} K)+1}, \quad \ell = 0, 1, 2, \ldots$$

where $K = \sum_{k=0}^{L} \frac{n^2}{4^k} = (4n^2 - 1)/3$ is the total number of search directions across all levels $1, 2, \ldots, L+1$ for this unconstrained optimisation problem. We first investigate

Algorithm 6 SC2 – Algorithm to solve the new primal-dual model

Given image z, an initial guess u, the stop tolerance (tol), and maximum multilevel cycle (maxit) with $L+1$ levels. Set $w = u$,

1) *Solve* (5.10) *to update u using the following steps:*

   i). *Set $\tilde{u} = u$.*

   ii). *Smooth for t iteration the approximation on the finest level 1 that is solve* (5.23) *for $i, j = 1, 2, ...n$*

   iii). *Iterate for t times on each coarse level $k = 2, 3, ...L, L+1$:*
   > If $k \leq L$, compute the minimiser c of (5.26)
   > Solve (5.27) *on the coarsest level $k = L+1$*
   > Add the correction $u = u + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b \times b$ block on level k as illustrated in (5.16).

2) *Solve* (5.11) *to update w:*

   i). *Set $\tilde{w} = w$.*

   ii). *Compute w using the formula* (5.9).

3) *Check for convergence using the above criteria. If not satisfied, return to Step 1. Otherwise exit with solution $u = \tilde{u}$ and $w = \tilde{w}$*

---

these search directions $\{\tilde{\mathbf{e}}\}$ and see that, noting $b_k = 2^{k-1},\ \tau = n/b_k$,

$$
\begin{array}{lll}
\text{level } k=1, & \tilde{\mathbf{e}}^j = e_j, & j = 1, 2, ..., n^2 \\
\text{level } k=2, & \tilde{\mathbf{e}}^{n^2+j} = e_{sj} + e_{sj+1} + e_{sj+n} + e_{sj+n+1} & j = 1, 2, ..., \frac{n^2}{4} \\
 & sj = b_k \left[\frac{j-1}{\tau_k}\right] n + \left(j - \tau \left[\frac{j-1}{\tau_k}\right] - 1\right) b_k + 1; & \\
\text{level } k=3, & \tilde{\mathbf{e}}^{n^2+\frac{n^2}{4}+j} = \sum_{\ell=0}^{3} \sum_{m=0}^{3} e_{sj+\ell n+m}, & j = 1, 2, ..., \frac{n^2}{4^2} \\
 & sj = b_k \left[\frac{j-1}{\tau_k}\right] n + \left(j - \tau \left[\frac{j-1}{\tau_k}\right] - 1\right) b_k + 1; & \\
\vdots & \vdots & \vdots \\
\text{level } k=L+1 & \tilde{\mathbf{e}}^K = \sum_{\ell=0}^{n-1} \sum_{m=0}^{n-1} e_{sj+\ell n+m} = \sum_{\ell=1}^{n^2} e_\ell, & j = \frac{n^2}{4^L} = 1 \\
 & sj = b_k \left[\frac{j-1}{\tau_k}\right] n + \left(j - \tau \left[\frac{j-1}{\tau_k}\right] - 1\right) b_k + 1 = 1; &
\end{array}
$$

where e_j denotes the j-th unit (coordinate) vector in $\mathbb{R}^{n^2}$, and on a general level k, with $\tau_k \times \tau_k$ pixels, the $j-$th index corresponds to position $(j - \tau_k[(j-1)/\tau_k],\ [(j-1)/\tau_k]+1)$ which is, on level 1, the global position $([(j-1)/\tau_k]b_k + 1,\ (j - \tau_k[(j-1)/\tau_k] - 1)b_k + 1)$ which defines the sum of unit vectors in a $b_k \times b_k$ block – see Figure 4.2(a-e). Clearly the sequence $\{p^\ell\}$ is essentially periodic (finitely many) and free-steering (spanning $\mathbb{R}^{n^2}$) [105].

Recall that a sequence $\{u^\ell\}$ is strongly downward (decaying) with respect to $g(u)$ i.e.

$$
g(u^\ell) \geq g(v^\ell) \geq g(u^{\ell+1}), \quad v^\ell = (1-t)u^\ell + tu^{\ell+1} \in D_0, \ \ \forall\ t \in [0, 1]. \tag{5.28}
$$

This property is much stronger than the usual decaying property $g(u^\ell) \geq g(u^{\ell+1})$ which is automatically satisfied by our Algorithm **SC2**.

By [105, Thm 14.2.7], to ensure the minimising sequence $\{u_\ell\}$ to be strongly downward, we modify the subproblem $\min J_1^{loc}(u^\ell + cp^\ell, c_1, c_2)$ to the following

$$u^{\ell+1} = u^\ell + c^* q^\ell, \qquad c^* = \text{argmin}\{c \geq 0 \mid \nabla J^T q^\ell = 0\}, \qquad \ell \geq 0 \tag{5.29}$$

where the ℓ-th search direction is modified to

$$q^\ell = \begin{cases} p^\ell, & \text{if } \nabla J^T p^\ell \leq 0, \\ -p^\ell, & \text{if } \nabla J^T p^\ell > 0. \end{cases}$$

Here the equation $\nabla J^T q^\ell = 0$ for c and the local minimising subproblem (5.24) i.e. $\min_c J_1^{loc}(\hat{u}_{i,j} + c, c_1, c_2)$ are equivalent. Now the new modification is to enforce $c \geq 0$ and the sequence $\{q^\ell\}$ is still essentially periodic.

We shall call the modified algorithm **SC2M**.

5.6 Convergence and complexity analysis

Proving convergence of the above algorithms **SC1-SC2** for

$$\min_{u \in \mathbb{R}} g(u)$$

would be a challenging task unless we make a much stronger assumption of uniform convexity for the minimising functional g. However it turns out that we can prove the convergence of **SC2M** for solving problem (5.22) without such an assumption. For theoretical purpose, we assume that the underlying functional $g = g(u)$ is hemivariate i.e. $g(u + t(v - u)) = g(u)$ for t in $[0, 1]$ and $u \neq v$.

To prove convergence of **SC2M**, we need to show that these 5 sufficient conditions are met

i) $g(u)$ is continuously differentiable in $D_0 = [0, 1]^{n^2} \subset \mathbb{R}^{n^2}$;

ii) the sequence $\{q^\ell\}$ is uniformly linearly independent;

iii) the sequence $\{u^\ell\}$ is strongly downward (decaying) with respect to $g(u)$;

iv) $\lim_{\ell \to \infty} g'(u^\ell) q^\ell / \|q^\ell\| = 0$,

v) the set $S = \{u \in D_0 \mid g'(u) = 0\}$ is non-empty.

Here $g'(u) = (\nabla g(u))^T$. Then we have the convergence of $\{u^\ell\}$ to a critical point u^* [105, Thm 14.1.4]

$$\lim_{\ell \to \infty} \inf_{u \in S} \|u^\ell - u^*\| = 0.$$

We now verify these conditions. Firstly condition i) is evident if $\beta \neq 0$ and condition ii) also holds since 'essentially periodic' implies 'uniformly linearly independent' [105,

§14.6.3]. Condition v) requires an assumption of existence of stationary points for $g(u)$. Below we focus on verifying iii)-iv). From [105, Thm 14.2.7], the construction of $\{u^\ell\}$ via (6.25) ensures that the sequence $\{u^\ell\}$ is strongly downward and further $\lim_{\ell\to\infty} g'(u^\ell)q^\ell/\|q^\ell\| = 0$. Hence conditions iii)-iv) are satisfied.

Note condition iii) and the assumption of $g(u)$ being hemivariate imply that $\lim_{\ell\to\infty} \|u^{\ell+1} - u^\ell\| = 0$ from [105, Thm 14.1.3]. Further condition iv) and the fact $\lim_{\ell\to\infty} \|u^{\ell+1} - u^\ell\| = 0$ lead to the result $\lim_{\ell\to\infty} g'(u^\ell) = \mathbf{0}$. Finally by [105, Thm 14.1.4], the condition $\lim_{\ell\to\infty} g'(u^\ell) = \mathbf{0}$ implies $\lim_{\ell\to\infty} \inf_{u\in S} \|u^\ell - u^*\| = 0$. Hence the convergence is proved.

Next, we will give the complexity analysis of our **SC1**, **SC2** and **SC2M**. Let $N = n^2$ be the total number of pixels (unknowns). First, we compute the number of floating point operations (flops) for **SC1** for level k as follows:

Quantities	Flop counts for **SC1**
$h,\ \upsilon$	$4b_k\tau_k^2$
θ term	$2N$
data term	$2N$
α term	$2N$
$s$smoothing steps	$38b_k\tau_k^2 s$

Then, the flop counts for all level is $W_{SC1} = \sum_{k=1}^{L+1} \left(6N + 4b_k\tau_k^2 + 38b_k\tau_k^2 s\right)$ where $k = 1$ (finest) and $k = L+1$ (coarsest). Noting $b_k = 2^{k-1}, \tau_k = n/b_k, N = n^2$, we compute the upper bound for **SC1** as follows:

$$W_{SC1} = 6(L+1)N + \sum_{k=1}^{L+1}\left(\frac{4N}{b_k} + \frac{38Ns}{b_k}\right) = 6(L+1)N + (4+38s)N\sum_{k=0}^{L}\left(\frac{1}{2^k}\right)$$
$$< 6N\log n + 14N + 76Ns \approx O\left(N\log N\right)$$

Similarly, the flops for **SC2** is given as

Quantities	Flop counts for **SC2**
$h,\ \upsilon$	$4b_k\tau_k^2$
ρterm	$2N$
w term	$6N$
$s$smoothing steps	$31b_k\tau_k^2 s$

Hence, the total flop counts for SC2 is $W_{SC2} = 6N + \sum_{k=1}^{L+1}\left(2N + 4b_k\tau_k^2 + 31b_k\tau_k^2 s\right)$. This gives the upper bound for **SC2** as

$$W_{SC2} = 6N + 2(L+1)N + \sum_{k=1}^{L+1}\left(\frac{4N}{b_k} + \frac{31Ns}{b_k}\right)$$
$$= 6N + 2(L+1)N + (4+31s)N\sum_{k=0}^{L}\left(\tfrac{1}{2^k}\right) < 2N\log n + 16N + 62Ns \approx O\left(N\log N\right)$$

Finally, the approximate cost of an extra operation $\nabla J^T q^\ell$ in **SC2M** is $2N$ that results to the total flop counts for SC2M as $W_{SC2M} = 6N + \sum_{k=1}^{L+1} \left(4N + 4b_k\tau_k^2 + 31b_k\tau_k^2 s\right)$. This gives the upper bound for **SC2M** as

$$
\begin{aligned}
W_{SC2M} &= 6N + 4(L+1)N + \sum_{k=1}^{L+1} \left(\frac{4N}{b_k} + \frac{31Ns}{b_k}\right) \\
&= 6N + 4(L+1)N + (4+31s)N \sum_{k=0}^{L} \left(\tfrac{1}{2^k}\right) < 4N\log n + 18N + 62Ns \approx O\left(N\log N\right)
\end{aligned}
$$

One can observe that both **SC1**, **SC2** and **SC2M** are of the optimal complexity $O(N\log N)$ expected of a multilevel method and $W_{SC1} > W_{SC2M} > W_{SC2}$.

5.7 Numerical experiments

This section will demonstrate the performance of the developed multilevel methods through several experiments. The algorithms to be compared are:

Name	**Algorithm**		**Description**
CMT	Old	:	The selective segmentation model proposed by Liu *et al.* [83] solved by a multilevel algorithm.
NCZZ	Old	:	The interactive image segmentation model proposed by Nguyen *et al.* [103] solved by a Split Bregman method.
BC	Old	:	The selective segmentation model proposed by Badshah and Chen [12] solved by an AOS algorithm.
RC	Old	:	The selective segmentation model proposed by Rada and Chen [111] solved by an AOS algorithm.
SC0	Old	:	The modified AOS algorithm [129] for the CDSS model [129].
SC1	New	:	The multilevel Algorithm 5 for the CDSS model [129].
SC2	New	:	The multilevel Algorithm 6 for the new primal-dual model (5.10)–(5.11).
SC2M	New	:	The modified multilevel algorithm for SC2.

There are five sets of tests carried out. In the **first set**, we will choose the best multilevel algorithm among SC1, SC2 and SC2M by comparing their segmentation performances in terms of CPU time (in seconds) and quality. The segmentation quality is measured based on the Jaccard similarity coefficient (JSC):

$$JSC = \frac{|S_n \cap S_*|}{|S_n \cup S_*|}$$

where S_n is the set of the segmented domain u and S_* is the true set of u (which is only easy to obtain for simple images). The similarity functions return values in the range $[0, 1]$. The value 1 indicates perfect segmentation quality while the value 0 indicates poor quality.

In the **second set**, we will perform the speed, quality, and parameter sensitivity test for the chosen multilevel algorithm (from set 1) and compare its performance with

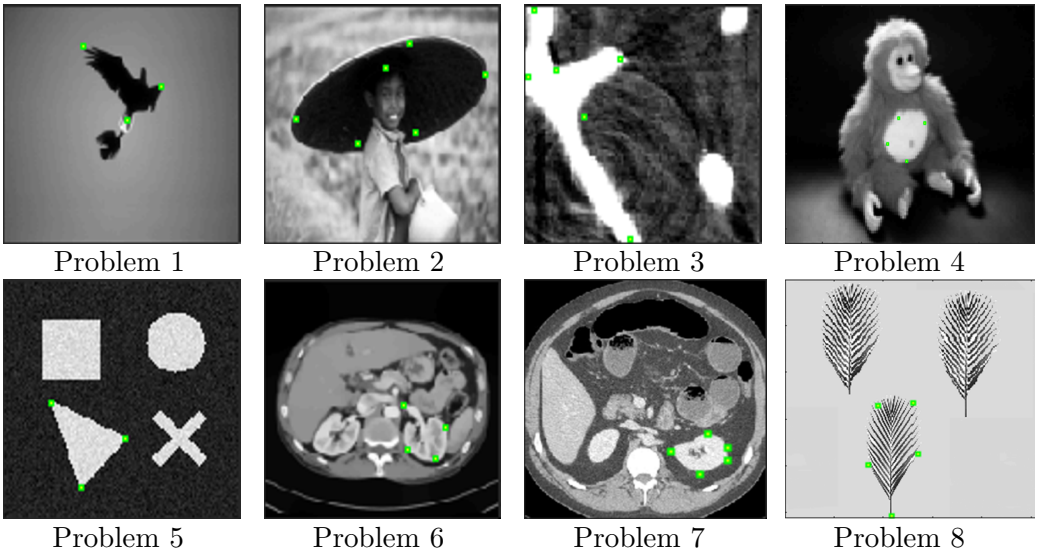


Figure 5.1: Segmentation test images and markers.

SC0. In the **third, fourth**, and **fifth set**, we will perform the segmentation quality comparison of the chosen multilevel algorithm (from set 1) with CMT model [83], NCZZ model [103], and BC model [12] and RC model [111] respectively.

The test images used in this paper are listed in Figure 5.1. We remark that Problems 1-2 are obtained from the Berkeley segmentation dataset and benchmark [92], while Problems 3-4 are obtain from database provided by [47]. All algorithms are implemented in MATLAB R2017a on a computer with Intel Core i7 processor, CPU 3.60GHz, 16 GB RAM CPU.

As a general guide to choose suitable parameters for different images, our experimental results recommend the following. The parameters $\bar{\mu} = \mu$ can be between 10^{-5} and 5×10^5, $\beta = 10^{-4}$, ρ in between 10^{-5} and 10^{-1}, and γ in between $1/255^2$ and 10. Tuning the parameter θ depends on the targeted object. If the object is too close to a nearby boundary then θ should be large. Segmenting a clearly separated object in an image needs just a small θ.

5.7.1 Test Set 1: Comparison of SC1, SC2, and SC2M

In the first experiment, we compare the segmentation speed and quality for SC1, SC2 and SC2M using test Problem 1-4 with size of 128×128. Here, we take $\bar{\mu} = 1$, $\beta = 10^{-4}$, $\rho = 10^{-3}$, $\theta = 1000$ (Problem 1-3), $\theta = 2000$ (Problem 4), $\varepsilon = 0.12$, $\gamma = 10$, $tol = 10^{-2}$ and $maxit = 10^4$.

Figure 5.2 shows successful selective segmentation results by SC1, SC2 and SC2M for Problem 4. The segmentation quality for all algorithms is the same (JSC=0.96). However, SC2 performs faster (4.9 seconds) than SC1 (10.5 seconds) and SC2M (6.3 seconds).

The remaining results are tabulated in Table 5.1. We can see for all four test

Table 5.1: Test Set 1 – Comparison of computation time (in seconds) and segmentation quality of SC1, SC2, and SC2M for Problem 1- 4. Clearly, for all four test problems, SC2 gives the highest accuracy and performs fast segmentation process compared to SC1 and SC2M.

Algorithm	Problem	Iteration	CPU time (s)	JSC
SC1	1	6	7.0	0.82
	2	12	20.0	0.82
	3	15	24.4	0.91
	4	6	10.5	0.96
SC2	1	5	5.9	0.82
	2	8	8.7	0.82
	3	4	4.9	0.91
	4	4	4.9	0.96
SC2M	1	5	7.9	0.79
	2	8	11.7	0.82
	3	5	7.9	0.85
	4	4	6.3	0.96

problems, SC2 gives the highest accuracy and performs the fastest compared to SC1 and SC2M.

Next, we test the performance of all the multilevel algorithms to segment Problem 5 in different resolutions. We take $\bar{\mu} = 1$, $\beta = 10^{-4}$, $\rho = 10^{-5}$, $\theta = 5000$, $\varepsilon = 0.12$, $\gamma = 10$, $tol = 10^{-3}$ and $maxit = 10^4$. The segmentation results for image size 1024×1024 are shown in Figure 5.3. The CPU times needed by SC2 to complete the segmentation of image size 1024×1024 is 413.2s while SC1 and SC2M need 690.6s and 636.1s respectively which implies that SC2 can be 277s faster than SC1 and 222s faster than SC2M. All the algorithms reach equal quality of segmentation.

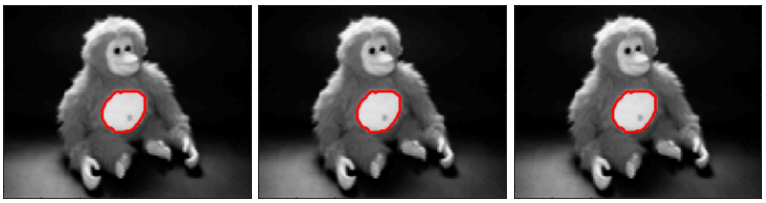
The remaining result in terms of quality and CPU time are tabulated in Table 5.2. Column 6 (ratios of the CPU times) shows that SC1, SC2 and SC2M are of complexity $O(N \log N)$. Again, we can see that for all image sizes, all algorithms have equal quality but SC2 is faster than other algorithms.

To illustrate the convergence of our multilvel algorithms, we plot in Figure 5.4 the residuals of SC1, SC2 and SC2M in segmenting Problem 5 for size 128×128 based on Table 5.2. There we extend the iterations up to 10. As we can see, the residuals of the algorithms keep reducing. The residuals for SC2 and SC2M decrease more rapidly than SC1.

Based on the experiments above, we observe that SC2 performs faster than the other two multilevel algorithms. In addition, for all problems tested, SC2 gives the higher segmentation quality than SC1 and SC2M. Therefore in practice, we recommend SC2 as the better multilevel algorithm for our convex selective segmentation method.

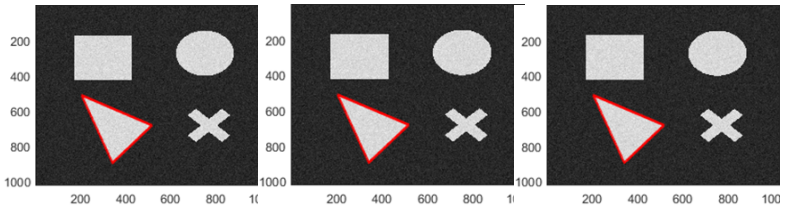
5.7.2 Test Set 2: Comparison of SC2 with SC0

The second set starts with the speed and quality comparison of SC2 with SC0 in segmenting Problem 5 with multiple resolutions. We take $\bar{\mu} = \mu = 1$, $\beta = 10^{-4}$,



SC1 SC2 SC2M

Figure 5.2: Test Set 1 – Segmentation of Problem 4 using our multilevel algorithms SC1, SC2, and SC2M with same quality (JSC=0.96) achieved. However, SC2 performs faster (4.9 seconds) compared to SC1 (10.5 seconds) and SC2M (6.3 seconds).

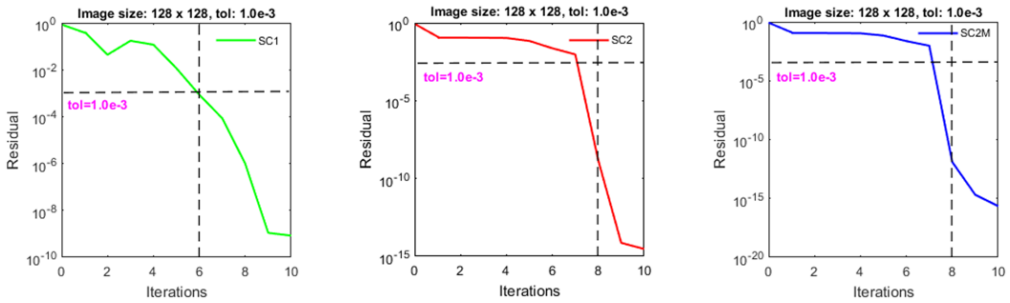


SC1 SC2 SC2M

Figure 5.3: Test Set 1 – Segmentation of Problem 5 of size 1024x1024 for SC1, SC2, and SC2M. SC2 can be 277 seconds faster than SC1 and 222 seconds faster than SC2M : see Table 5.2. All algorithms give similar segmentation quality.

Table 5.2: Test Set 1 – Comparison of computation time (in seconds) and segmentation quality of SC1, SC2 and SC2M for Problem 5. The time ratio, t_n/t_{n-1} close to 4.4 indicates $O(N \log N)$ speed. Clearly, all algorithms have similar quality but SC2 is faster than SC1 and SC2M for all image sizes.

Algorithm	Size $N = n \times n$	Unknowns N	Iteration	Time, t_n	$\frac{t_n}{t_{n-1}}$	JSC
SC1	128×128	16384	6	10.6		1.0
	256×256	65536	7	43.5	4.1	1.0
	512×512	262144	7	173.7	4.0	1.0
	1024×1024	1048576	7	690.6	4.0	1.0
SC2	128×128	16384	8	8.7		1.0
	256×256	65536	7	23.7	2.7	1.0
	512×512	262144	8	103.9	4.4	1.0
	1024×1024	1048576	8	413.2	4.0	1.0
SC2M	128×128	16384	8	11.6		1.0
	256×256	65536	7	36.5	3.1	1.0
	512×512	262144	8	156.7	4.3	1.0
	1024×1024	1048576	8	636.1	4.1	1.0



SC1 SC2 SC2M

Figure 5.4: Test Set 1 – The residual plots for SC1, SC2, and SC2M to illustrate the convergence of the algorithms. The extension up to 10 iterations shows that the residual of the algorithms keep reducing. The residual for SC2 and SC2M decrease rapidly compared to SC1.

Table 5.3: Test Set 2 – Comparison of computation time (in seconds) and segmentation quality of SC0 and SC2 for Problem 5 with different resolutions. Again, the time ratio, $t_n/t_{n-1} \approx 4.4$ indicates $O(N \log N)$ speed since $N_L = n_L^2 = (2^L)^2 = 4^L$ and $kN_L \log N_L/(kN_{L-1} \log N_{L-1}) = 4L/(L-1) \approx 4.4$. Clearly, all algorithms have similar quality but SC2 is faster than SC0 for all image sizes. Here, (**) means taking too long to run. For image size 512×512, SC2 performs 33 times faster than SC0.

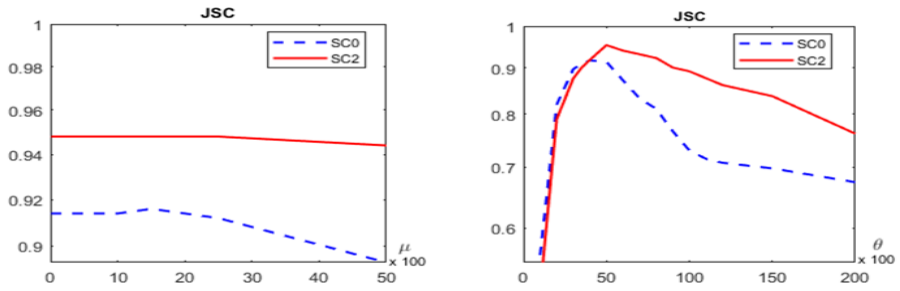
Algorithm	Size $N = n \times n$	Time, t_n	$\frac{t_n}{t_{n-1}}$	JSC
SC0	128×128	243.5		1.0
	256×256	872.7	3.6	1.0
	512×512	3803.1	4.4	1.0
	1024×1024	**	**	**
SC2	128×128	8.6		1.0
	256×256	27.2	3.2	1.0
	512×512	112.0	4.1	1.0
	1024×1024	453.6	4.1	1.0

$\rho = 10^{-5}$, $\theta = 5000$, $\varepsilon = 0.01$, $\gamma = 10$, $tol = 10^{-6}$ and $maxit = 5000$.

The segmentation results are tabulated in Table 5.3. The ratios of the CPU times in column 4 show that SC0 and SC1 are of complexity $O(N \log N)$. The symbols (**) indicates that too much time is taken to complete the segmentation task. For all image sizes, SC0 and SC2 give the same high quality.

Next, we shall test parameter sensitivity for our recommended SC2. We focus on three important parameters: the regularisation parameter μ, the regularising parameter β and the area parameter θ. The SC2 results are compared with SC0.

Test on parameter μ. The regularisation parameter μ in a segmentation model not only controls a balance of the terms but also implicitly defines the minimal diameter of detected objects among a possibly noisy background [152]. Here, we test sensitivity of SC2 for different regularisation parameters μ in segmenting an object in Problem 6 and compare with SC0 in terms of segmentation quality. We set $\beta = 10^{-4}$, $\rho = 10^{-5}$, $\varepsilon = 0.01$, $\gamma = 1/255^2$, $\theta = 5000$, $tol = 10^{-5}$ and $maxit = 10^4$.



(a) (b)

Figure 5.5: Test Set 2 – The segmentation accuracy for SC0 and SC2 in segmenting Problem 6 using different values of parameter μ in (a) and parameter θ in (b). The results demonstrate that SC2 is successful for a much larger range for both parameters.

Table 5.4: Test Set 2 – Dependence of our SC2 on β for segmenting Problem 6 in Figure 4.4.

β	JSC	Energy
1	0.95	-5.342264e+06
10^{-1}	0.95	-5.341634e+06
10^{-5}	0.95	-5.342115e+06
10^{-10}	0.95	-5.342146e+06
10^{-15}	0.95	-5.342102e+06

Figure 5.5a shows the value of JSC for SC0 and SC2 respectively for different values of μ. Clearly, SC2 is successful for larger range of μ than SC0. This finding implies that SC2 is less dependent to parameter μ than SC0.

Test on area parameter θ. As a final comparison of SC0 and SC2, we will test how the area parameter θ effects the segmentation quality of SC0 and SC2. For this comparison, we use Problem 6 and set $\bar{\mu} = \mu = 100$, $\beta = 10^{-4}$, $\rho = 10^{-3}$, $\varepsilon = 0.01$, $\gamma = 1/255^2$, $tol = 10^{-5}$ and $maxit = 10^4$. Figure 5.5b shows the value of JSC for SC0 and SC2 respectively for different values of θ. We observe that SC2 is successful for a larger range of θ than SC0. This finding implies that SC2 is less sensitive to parameter θ than SC0.

Test on parameter β. Finally, we examine the sensitivity of our proposed SC2 on parameter β. The parameter β is used to avoid singularity or to ensure the original cost function is differentiable and it should be as small as possible (close to 0) so that the modified cost function (having β) in (5.22) is close to the original cost function in (5.10). We have chosen to segment an object (organ) in Problem 6. Six different values of β are tested: $\beta = 1$, 10^{-1}, 10^{-5}, 10^{-10}, and 10^{-15}. Here, $\bar{\mu} = 100$, $\rho = 10^{-3}$, $\theta = 5500$, $\gamma = 1/255^2$, $tol = 10^{-3}$ and $maxit = 10^4$. For quantitative analysis, we compute the energy value in equation (5.10) (that has no β) and the JSC value. Both values are tabulated in Table 5.4. One can see that as β decreases, the energy value gets closer to each other. The segmentation quality measured by JSC values remain the same as β decreases. This result indicates that SC2 is not very sensitive to β.

5.7.3 Test Set 3: Comparison of SC2 with CMT model [83]

The CMT and SC2 models contain two stages. The first stage is to get an approximation solution aided by the markers set. The second stage is to use the approximation solution and perform a thresholding procedure to obtain the object of interest in binary representation. In this test set 3, we investigate how the number of markers and threshold values will effect the segmentation quality for CMT model [83] and our SC2. For this purpose, we use the test Problem 4. We set $\bar{\mu} = 10^{-5}$, $\beta = 10^{-4}$, $\rho = 20$, $\theta = 3.5$, $\gamma = 20$, $tol = 10^{-3}$ and $maxit = 10^4$. The first row in Figure 5.6 shows the Problem 4 with different number of markers. There are 4 markers in (a1), 6 markers in (b1) and 9 markers used in (c1). The results given by CMT and SC2 using the markers with different threshold value are plotted respectively in the second row.

We observe that CMT performs well only when the number of markers used is large while our SC2 is less sensitive to the number of markers used. In addition, it is clearly shown that the range of threshold values that work for SC2 is wider than CMT.

To explain this, we can see the formulation of CMT and SC2 models. In one hand, the minimisation of functional (5.2) of CMT model in the first stage gives a piecewise smooth intensities approximation of the targeted object. As result, a suitable range of threshold value in the second stage can be small. One can increase the number of markers used in CMT model to get a good approximation, hence increase the suitable range of the threshold value. On the other hand, the formulation of SC2 in (5.10) and (5.11) assumes that an image comprises of two regions of approximately piecewise constant intensities of distinct values c_1 and c_2, separated by some contour Γ. The object to be detected (foreground) is represented by the the value c_1 and the value c_2 represents the background. This assumption allows a wide suitable range of the threshold value for SC2 and less marker set is needed compared to CMT model.

We remark that another possible comparison is to compute the area under the curves (a2), (b2), and (c2). The higher the area under the curve, the less dependent the method towards number of markers.

5.7.4 Test Set 4: Comparison of SC2 with NCZZ model [103]

The NCZZ model uses two types of markers to label foreground and background region while SC2 uses only one marker set to label foreground region. To avoid bias, we ensure that the foreground markers for both models are placed inside or as near as possible to the targeted object while the background marker for NCZZ is placed outside the targeted object. For almost all of the test images in Figure 5.1, we see that the NCZZ model [103] gives same satisfactory results as our SC2. For brevity, we will not show too many cases where both models give satisfactory results; Figure 5.7 shows the successful segmentation of an organ in Problem 7 of size 256×256 by NCZZ model. The markers used to label foreground region (red) and background region (blue) for the NCZZ model [103] are shown in Figure 5.7(a) and a markers set used to label foreground for SC2 is shown in Figure 5.1. Successful segmentation results (zoom in) by NCZZ model [103] and our SC2 for Problem 7 are shown in Figure 5.7(b) and 5.7(c) respectively using

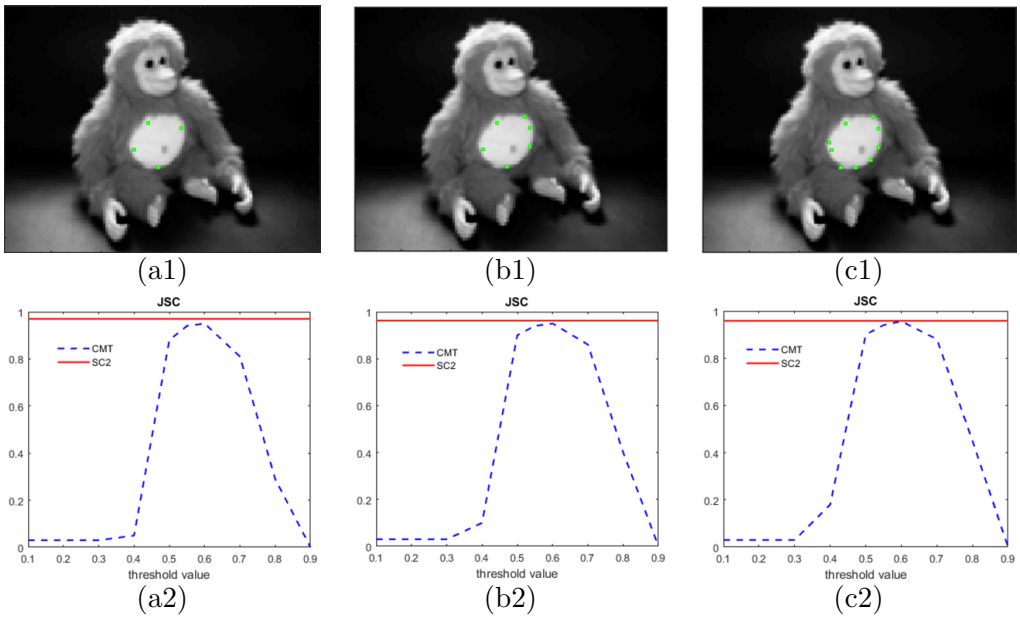
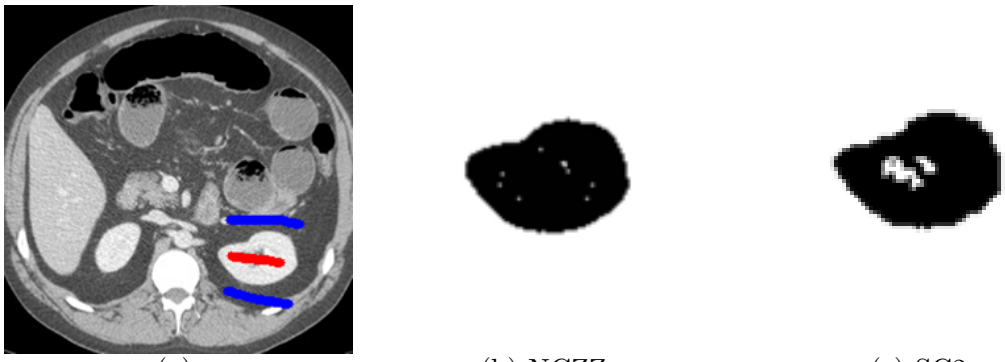


Figure 5.6: Test Set 3 – Comparison of SC2 with CMT model [83]. First row shows different numbers of markers used for Problem 4. Second row demonstrates the respective results (a2), (b2) and (c2) for (a1), (b1) and (c1) with different threshold values. Clearly, CMT performs well only when the number of markers used is large while our SC2 seems less sensitive to the number of markers used. Furthermore, the range of threshold value that works for SC2 is wider than CMT.

the following parameters; $\bar{\mu} = 0.01$, $\beta = 10^{-4}$, $\rho = 10^{-3}$, $\theta = 3000$, $\gamma = 10$, $tol = 10^{-2}$ and $maxit = 10^4$. However, we can observe that our SC2 gives more details of features inside the organ compare to NCZZ model. This extra information may be beneficial to the medical practitioners for further diagnostic process.

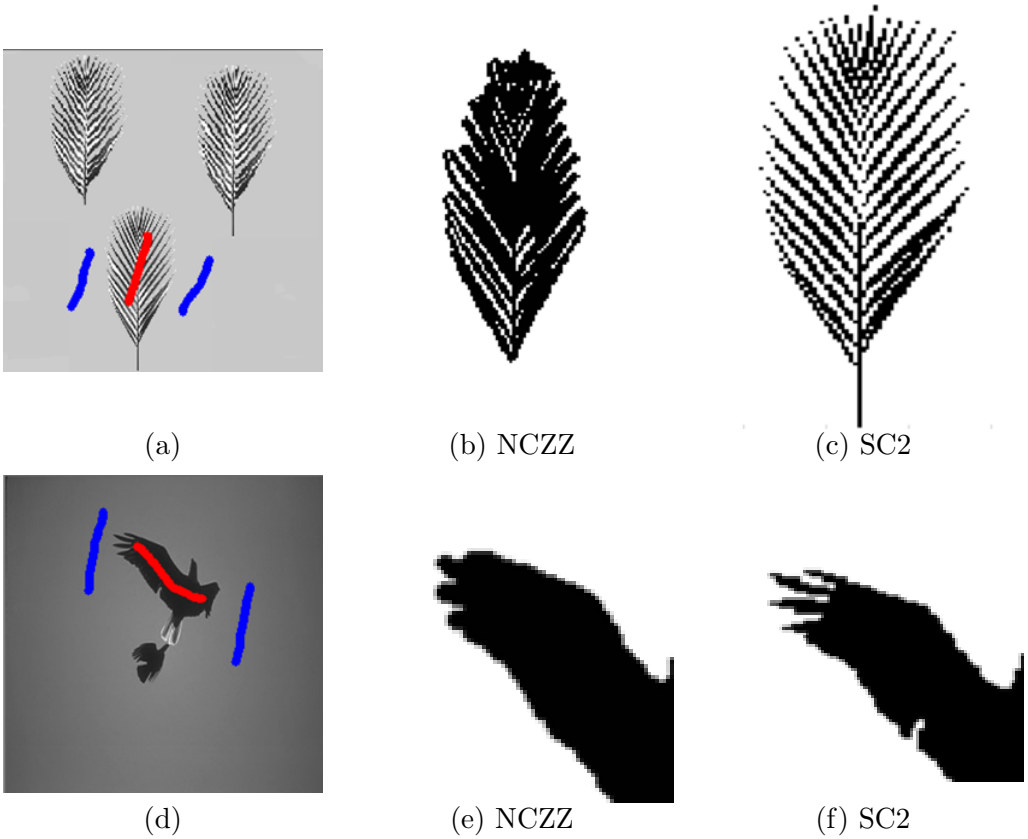
However, according to the authors [103], the model unable to segment semi-transparent boundaries and sophisticated shapes (such as bush branches or hair in a clean way due to its underlying assumption that the shape of the object is smooth and can be well described by the weighted shortest boundary length and as a hard segmentation, a pixel is assigned to only one class [103]. In Figure 5.8, we demonstrate the limitation of NCZZ model using Problems 1 and 8. The set of parameters are $\bar{\mu} = 0.01$, $\beta = 10^{-4}$, $\rho = 10^{-3}$, $\theta = 2000$ (Figure 5.8(a)), $\theta = 400$ (Figure 5.8(d)), $\gamma = 10$, $tol = 10^{-2}$ and $maxit = 10^4$. The markers used to label foreground region (red) and background region (blue) for the NCZZ model [103] for Problem 1 and 8 are shown in Figure 5.8(a) and 5.8(d), respectively. A markers set used in SC2 to label foreground of Problem 1 and 8 are shown in Figure 5.1.

Zoomed segmentation results in Figure 5.8(b) and (e) demonstrate the limitation of NCZZ model [103]. As comparison, our SC2 gives cleaner segmentation as illustrated in Figure 5.8(c) and (f) for the same problems. We remark that SC2 is also a hard segmentation method, however the Euclidean distance from polygon region, Q used in the formulation of SC2 is helpful to provide a good estimation as the function allows the solution to be constrained by values associated with Q, associated with the marker set A.



(a) (b) NCZZ (c) SC2

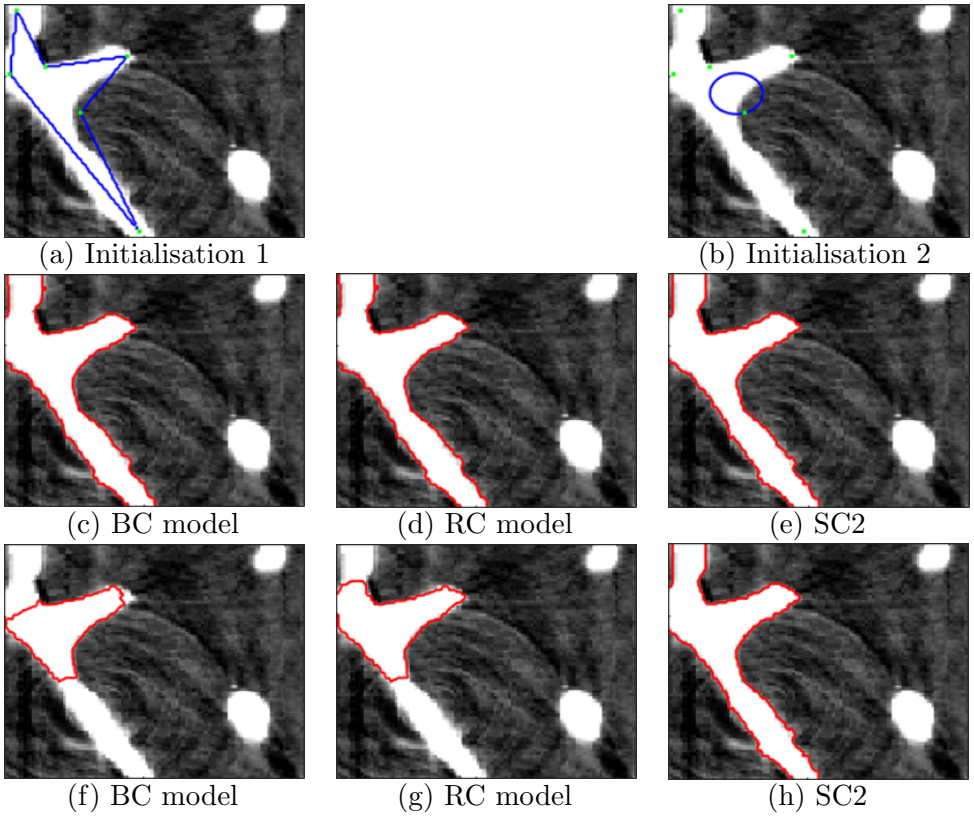
Figure 5.7: Problem 7 in Test Set 4 – Two types of markers used to label foreground region (red) and background region (blue) for NCZZ model [103] in (a). Successful segmentation result (zoom in): (b) by NCZZ model [103] and (c) by our SC2 (only using foreground markers).



(a) (b) NCZZ (c) SC2

(d) (e) NCZZ (f) SC2

Figure 5.8: Problems 1,8 in Test Set 4 – (a) and (d) show the foreground markers (red) and background markers (blue) for NCZZ model [103]. Zoomed segmentation results in (b) and (e) demonstrate the limitation of NCZZ model [103] that is unable to segment semi-transparent boundaries and sophisticated shapes (such as bush branches or hair as explained in [103]) in a clean way. Our SC2 gives cleaner segmentation for the same problems as illustrated in (c) and (f).



(a) Initialisation 1 (b) Initialisation 2

(c) BC model (d) RC model (e) SC2

(f) BC model (g) RC model (h) SC2

Figure 5.9: Test Set 5 – Performance comparison of BC, RC and SC2 models using 2 different initialisations. With Initialisation 1 in (a), the segmentation results for BC, RC, and SC2 models are illustrated on second row (c-e) respectively. With Initialisation 2 in (b), the results are shown on third row (f-h). Clearly, SC2 gives a consistent segmentation result indicating that our SC2 is independent of initialisations while BC and RC are sensitive to initialisations due to different results obtained.

5.7.5 Test Set 5: Comparison of SC2 with BC [12] and RC [111]

Finally, we compare the performance of SC2 with two non-convex models namely BC model [12] and RC model [111] for different initialisations in segmenting Problem 3. We set $\bar{\mu} = 128 \times 128 \times 0.05$, $\beta = 10^{-4}$, $\rho = 10^{-4}$, $\theta = 1000$, $\gamma = 5$, $tol = 10^{-4}$ and $maxit = 10^4$. Figures 5.9(a) and 5.9(b) show two different initialisations with fixed markers.

The second row shows the results for all three models using the first initialisation in (a) and the third row using the second initialisation in (b). It can be seen that under different initialisations, our SC2 will result in the same, consistent segmentation curves (hence independent of initialisations) showing the advantage of a convex model. However, the segmentation results for BC and RC models are heavily dependent on the initialisation; a well known drawback of non-convex models. In addition, the segmentation result of non-convex models is not guaranteed to be a global solution.

5.8 Summary

In this chapter, we presented a new primal-dual formulation for CDSS model [129] and proposed an optimisation based multilevel algorithm SC2 to solve the new formulation. In order to get a stronger decaying property than SC2, a new variant of SC2 named as SC2M is proposed. We also have developed a multilevel algorithm for the original CDSS model [129] called as SC1. In **Test Set 1** of the experiment, we found that all the multilevel algorithms having the expected optimal complexity $O(N \log N)$. However, SC2 converges faster than SC1 and SC2M. In addition, for all tested images, SC2 gives high accuracy compared to SC1 and SC2M. Practically, we recommended SC2 as the better multilevel algorithm for convex and selective segmentation method. In **Test Set 2**, we have performed the speed and quality comparisons of SC2 with SC0. Results show that SC2 performs much faster compared to SC0. Both algorithms deliver same high quality for the tested problem. We also have run the sensitivity test for our recommended algorithm SC2 towards parameters μ and θ. Comparison of SC2 with SC0 shows that SC2 is less sensitive to the regularisation parameters μ and θ. Moreover, SC2 also less sensitive for parameter β. In **Test Set 3**, we compare the segmentation quality of SC2 with a recent model namely CMT. The result demonstrates that SC2 perform better than CMT even for few markers. Moreover, the range of threshold value works for SC2 is wider than CMT. In **Test Set 4**, the segmentation quality of SC2 is compared with NCZZ model. For the tested problem, it is clearly that SC2 is successfully reduce the difficulty of NCZZ model that is unable to segment semi-transparent boundaries and sophisticated shapes. The final **Test Set 5** demonstrate the advantage of SC2 being a convex model (independent of initialisation) compare to two non convex models (BC and RC). We will extend SC2 to 3D formulation and develop an optimisation based multilevel algorithm in 3-D framework in the next chapter.

Chapter 6

A Three-dimensional Convex and Selective Variational Image Segmentation Model and Its Fast Algorithms

This chapter presents a new three-dimensional (3-D) convex selective segmentation model. In order to process 3-D images of large size associated with high resolution, an optimisation based multilevel algorithm in 3-D framework are proposed. This will be an extension of the 2-D model and 2-D multilevel algorithm already discussed in Chapter 5. Numerical tests show that the proposed model is effective and the algorithm is efficient in locally segmenting 3-D complex image structures.

6.1 Introduction

Various models, algorithms and techniques in 2-D may be generalised to the 3-D case, e.g. methods based on 2-D level set active contours may be generalised to 3-D level set active surfaces [39, 146, 151]. These methods have been widely used and successful for many applications where all features (objects) in a given image have to be segmented. However, as shown in Chapter 4 and 5 there are some other applications where the selection of one feature among many is required. This kind of problem leads to a new and challenging task of selective segmentation. Many application fields such as medical imaging, geological surveying and computational fluid dynamics can greatly benefit from 3-D selective segmentation, however there exist only a few works for selective segmentation in 3-D.

Some effective selective segmentation 3-D models are 3-D version of [12] that is implemented in [113] and 3-D selective model by [113]. A recent 3-D work by [150] applied the narrow band idea of [96] into [152]. All the 3-D models mentioned are non-convex. A 3-D non-convex selective variational image segmentation model, though effective in capturing a local minimiser, is sensitive to initialisation where the

segmentation result relies heavily on user input.

A recent 2-D convex selective models is introduced by [129] named CDSS, allowing a global minimiser to be found independently of initialisation. However, the model is sensitive to regularisation and area parameters. A stabilised version of the model named SC2 that is less sensitive to the parameters is introduced in [74]. In this chapter we will generalise the 2-D SC2 model [74] into 3-D formulation.

Another class of fast algorithm in 2-D framework is called Chambolle's projection algorithm [26]. This popular algorithm is considered as powerful [33] and fast method [20, 26, 40]. This is mainly because the algorithm able to solve the original convex variational functional (non-differentiable) without introducing parameter β (to avoid singularity). In 2-D global and convex variational image segmentation problem, the algorithm is used by D. Chen et al. [40] to solve a variant of Mumford-Shah model which handles the segmentation of medical images with intensity inhomogeneities and also in Moreno et al. [99] for solving a four phase model for segmentation of brain MRI images by active contours.

We will solve the new 3-D formulation of the SC2 model [74] using a new developed 3-D multilevel algorithm. Besides, we will apply Chambolle's projection algorithm in 3-D framework to solve the 3-D model.

The rest of the chapter is organised as follows. We shall first review some 3-D selective segmentation models in Section 6.2. Then in Section 6.3 we develop a 3-D formulation of SC2 model [74]. In Section 6.4, we present the 3-D optimisation based multilevel algorithm to solve the 3-D version of SC2 model. To speed up the convergence, we proposed a new localised version of 3-D SC2 model and solved using multilevel algorithm in Section 6.5. New variants of multilevel algorithms are presented in Section 6.6 and we discuss their convergence in Section 6.7. Section 6.8 shows experimental results. The last Section 6.9 concludes the chapter.

6.2 Review of existing 3-D segmentation models

In this section we will first introduce the extension of 2-D global segmentation model [38] into 3-D version by [151, 113] because it provides the foundation for the selective segmentation models as well as a method for minimising the associated functional. Next, we will discuss a recent 3-D selective segmentation models by Zhang-Chen [150] before we address the non convexity issue for these models.

6.2.1 The 3-D Chan-Vese model

The Chan and Vese (CV) model [38] considers a special case of the piecewise constant Mumford-Shah functional [102] where it is restricted to only two phases (i.e. constants), representing the foreground and the background of the given image $z(x, y)$. We now review the CV model [38] in the 3-D framework.

Let $\mathbf{x} = (x, y, z)$, hence define a given 3-D grey level image as $z(\mathbf{x}) : \Omega \subset \mathbb{R}^3 \to \mathbb{R}$. Assume that the 3-D image z is composed by two 3-D regions of approximately

piecewise constant intensities of distinct (unknown) values c_1 and c_2 , separated by some (unknown) 2-D surface Γ. Let the object to be detected be represented by the 3-D region Ω_1 with the value c_1 inside the 2-D surface Γ whereas outside Γ, in $\Omega_2 = \Omega \backslash \Omega_1$, the intensity of z is approximated with the value c_2. Then, with $\Omega = \Omega_1 \cup \Omega_2$, the Chan-Vese model minimises the following functional

$$\begin{aligned} \min_{\Gamma, c_1, c_2} F_{CV}^{3D}(\Gamma, c_1, c_2) = \mu \text{ surface area}(\Gamma) + \lambda_1 \int_{\Omega_1} (z(\mathbf{x}) - c_1)^2 d\mathbf{x} \\ + \lambda_2 \int_{\Omega_2} (z(\mathbf{x}) - c_2)^2 d\mathbf{x}. \end{aligned} \tag{6.1}$$

Here, the constants c_1 and c_2 are viewed as the average intensities values of z inside and outside the Γ. The fixed parameters μ, λ_1, and λ_2 are non-negative but need to be specified. In order to minimise equation (6.1), the level set method [38] is applied, where the unknown surface Γ is represented by the zero level set of the Lipschitz function such that

$$\begin{aligned} &\Gamma = \{(x, y, z) \in \Omega : \phi(x, y, z) = 0\}, \\ &\Omega_1 = \text{inside}(\Gamma) = \{(x, y, z) \in \Omega : \phi(x, y, z) > 0\}, \\ &\Omega_2 = \text{outside}(\Gamma) = \{(x, y, z) \in \Omega : \phi(x, y, z) < 0\}. \end{aligned}$$

To simplify the notation, denote the regularised versions of the Heaviside function and the Dirac delta function, respectively, by

$$H(\phi(x, y, z)) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\phi}{\varepsilon}\right)\right) \qquad \text{and} \qquad \delta(\phi(x, y, z)) = \frac{\varepsilon}{\pi(\varepsilon^2 + \phi^2)}.$$

Thus equation (6.1) becomes

$$\begin{aligned} &\min_{\phi, c_1, c_2} F_{CV}^{3D}(\phi, c_1, c_2) = \mu \int_{\Omega} |\nabla H(\phi)| \, d\mathbf{x} + \lambda_1 \int_{\Omega} (z(\mathbf{x}) - c_1)^2 H(\phi) \, d\mathbf{x} \\ &+ \lambda_2 \int_{\Omega} (z(\mathbf{x}) - c_2)^2 (1 - H(\phi)) \, d\mathbf{x}. \end{aligned} \tag{6.2}$$

Keeping the level set function ϕ fixed and minimising (6.2) with respect to c_1 and c_2, we have

$$c_1(\phi) = \frac{\int_{\Omega} z(\mathbf{x}) H(\phi) \, d\mathbf{x}}{\int_{\Omega} H(\phi) \, d\mathbf{x}}, \qquad c_2(\phi) = \frac{\int_{\Omega} z(\mathbf{x}) (1 - H(\phi)) \, d\mathbf{x}}{\int_{\Omega} (1 - H(\phi)) \, d\mathbf{x}}. \tag{6.3}$$

After that, by fixing constants c_1 and c_2 in $F_{CV}^{3D}(\phi, c_1, c_2)$, first variation with respect to ϕ yields the following Euler-Lagrange equation:

$$\begin{cases} \mu\delta(\phi) \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda_1 \delta(\phi)(z - c_1)^2 + \lambda_2 \delta(\phi)(z - c_2)^2 = 0, \quad in \quad \Omega \\ \frac{\delta(\phi)}{|\nabla\phi|} \frac{\partial u}{\partial \vec{n}} = 0, \quad on \quad \partial\Omega. \end{cases} \tag{6.4}$$

Notice that the nonlinear coefficient in equation (6.4) may have a zero denominator, so the equation is not defined in such cases. A commonly-adopted idea to deal with

$|\nabla\phi| = 0$ was to introduce a small positive parameter β to (6.2) and (6.4), so the new Euler Lagrange equation becomes

$$\begin{cases} \mu\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right) - \lambda_1\delta(\phi)(z-c_1)^2 + \lambda_2\delta(\phi)(z-c_2)^2 = 0, \quad in \quad \Omega \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \frac{\delta(\phi)}{|\nabla\phi|}\frac{\partial u}{\partial\vec{n}} = 0, \quad on \quad \partial\Omega. \end{cases}$$

where corresponds to minimising the following differentiable energy function, instead of (6.2)

$$\begin{aligned} &\min_{\phi,c_1,c_2} F_{CV}^{3D}(\phi,c_1,c_2) = \mu\int_\Omega \sqrt{|\nabla H(\phi)|^2+\beta}\,d\mathbf{x} \\ &+\lambda_1\int_\Omega (z-c_1)^2 H(\phi)\,d\mathbf{x} + \lambda_2\int_\Omega (z-c_2)^2(1-H(\phi))\,d\mathbf{x}. \end{aligned} \tag{6.5}$$

It should be remarked that applying a global segmentation model first and selecting an object next amount provide an alternative to selective segmentation. However this approach would require a secondary binary segmentation and is not reliable because the first round of segmentation cannot guarantee to isolate the interested object often due to non-convexity of models. Thus, urgent need exist in developing selective segmentation model.

6.2.2 The 3-D Zhang-Chen-Gould model

The 3-D selective segmentation model by Zhang-Chen-Gould (3DZCG) [150] improves the selective model [12] which combines the edge based model of [65, 67] with intensity fitting term similar to 3-D CV [39]. On 3-D image $z(\mathbf{x})$ defined on the cubic domain Ω, the selective segmentation idea can be described as the detection of the boundary of a single target object among all homogeneity intensity object that are defined in a closed domain to the geometrical points in a set $\mathrm{A} = \{w_i = (x_i^*, y_i^*, z_i^*) \in \Omega, 1 \le i \le n_1\} \subset \Omega$, consisting of $n_1\,(\ge 3)$ points on or near the target object [12]. The marker set or the geometrical points in the set A can be used to define an initial solution and to guide its evolution towards Γ. The 3DZCG model takes the form

$$\min_{\Gamma,c_1,c_2}\{E(\Gamma,c_1,c_2) = \mu E_G(\Gamma) + E_F(\Gamma,c_1,c_2)\} \tag{6.6}$$

where $E_G(\Gamma) = \int_\Gamma G(\mathbf{x})\,ds$ with $G(\mathbf{x}) = g(\mathbf{x})\,d(\mathbf{x})$. Here the function $g(\mathbf{x}) = \frac{1}{1+q|\nabla z(\mathbf{x})|^2}$ is an edge detector function which helps to stop the evolving curve on the edge of the targeted object in an image. The strength of detection is adjusted by a parameter q. The function $g(\mathbf{x})$ is constructed to take small values near to 0 near object edge and large values near to 1 in flat region. The function $d(\mathbf{x})$ is a marker distance function, which is close to 0 when approaching the points from marker set A given as:

$$d(\mathbf{x}) = \text{distance}((\mathbf{x}),\mathrm{A}) = \prod_{i=1}^{n_1}\left(1 - e^{-\frac{(\mathbf{x}-_i^*)^2}{2\kappa^2}}\right), \quad \forall(\mathbf{x}) \in \Omega$$

where κ is a positive constant. Alternative distance functions $d(\mathbf{x})$ are also possible [150]. The new local fitting energy $E_F(\Gamma, c_1, c_2)$ is

$$\left\{ \lambda_1 \int_{\Omega_{in}(\Gamma)} b_1(\phi(\mathbf{x}), \gamma_{in}) (z(\mathbf{x}) - c_1)^2 d\mathbf{x} + \lambda_2 \int_{\Omega_{out}(\Gamma)} b_2(\phi(\mathbf{x}), \gamma_{out}) (z(\mathbf{x}) - c_2)^2 d\mathbf{x} \right\}$$

Where $b_1(\phi(\mathbf{x}), \gamma_{in}) = B(\phi(\mathbf{x}), \gamma_{in}, 0)$ and $b_2(\phi(\mathbf{x}), \gamma_{out}) = B(\phi(\mathbf{x}), 0, \gamma_{out})$ with

$$B(\phi(\mathbf{x}), \gamma_{in}, \gamma_{out}) = H(\phi(\mathbf{x}) + \gamma_{in})(1 - H(\phi(\mathbf{x}) - \gamma_{out})) \tag{6.7}$$

characterizes the domain $\Omega_{\gamma_{in},\gamma_{out}} = \{\mathbf{x} \in \Omega : -\gamma_{in} \leq \phi(\mathbf{x}) \leq \gamma_{out}\} = \Omega_{\gamma_{in}}(\Gamma) \cup \Gamma \cup \Omega_{\gamma_{out}}(\Gamma)$ which is a narrow band region surrounding the local boundary Γ. They assume that ϕ is negative inside Γ and positive outside it. Using the level set formulation, the model is rewritten as:

$$\begin{aligned} &\min_{\phi, c_1, c_2} \mu \int_\Omega G(\mathbf{x}) |\nabla H(\phi)| \, d\mathbf{x} + \lambda_1 \int_\Omega (z - c_1)^2 b_1(\phi(\mathbf{x}), \gamma_{in})(1 - H(\phi)) \, d\mathbf{x} \\ &+ \lambda_2 \int_\Omega (z - c_2)^2 b_2(\phi(\mathbf{x}), \gamma_{out}) H(\phi) \, d\mathbf{x} \end{aligned} \tag{6.8}$$

Note that $b_1(\phi(\mathbf{x}), \gamma_{in})(1 - H(\phi)) = B(\phi(\mathbf{x}), \gamma_{in}, 0)$ and $b_2(\phi(\mathbf{x}), \gamma_{out})(H(\phi)) = B(\phi(\mathbf{x}), 0, \gamma_{out})$. Keeping the level set function ϕ fixed and minimising (6.8) with respect to c_1 and c_2, we have

$$c_1(\phi) = \frac{\int_\Omega z(\mathbf{x})(1 - H(\phi)) b_1 \, d\mathbf{x}}{\int_\Omega (1 - H(\phi)) b_1 \, d\mathbf{x}}, \qquad c_2(\phi) = \frac{\int_\Omega z(\mathbf{x}) H(\phi) b_2 \, d\mathbf{x}}{\int_\Omega H(\phi) b_2 \, d\mathbf{x}}$$

Finally keeping constants c_1 and $c_2$fixed yields the following Euler-Lagrange equation

$$\begin{cases} \mu\delta\nabla \cdot G\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda_1 \left[b_1\delta - (1 - H)\frac{\partial b_1}{\partial\phi}\right](z - c_1)^2 + \lambda_2 \left[b_2\delta + H\frac{\partial b_2}{\partial\phi}\right](z - c_2)^2 = 0, \; in\, \Omega \\ G\frac{\delta}{|\nabla\phi|} \frac{\partial\phi}{\partial\vec{n}} = 0, \quad on \quad \Omega \end{cases}$$

The above PDE is solved numerically by a multigrid method.

6.3 A 3-D convex and selective segmentation model

Recently, we have developed a new stabilised version of 2-D convex selective segmentation model [129] through primal-dual formulation called SC2 [74].

First, using the set A, construct a polygon Q that connects up the markers. Denote the function $P_d(\mathbf{x})$ as the Euclidean distance of each point $(\mathbf{x}) \in \Omega$ from its nearest point $(x_p, y_p, z_p) \in Q$:

$$P_d(\mathbf{x}) = \sqrt{(x - x_p)^2 + (y - y_p)^2 + (z - z_p)^2} = \min_{q \in Q} \|(x, y, z) - (x_q, y_q, z_q).\|$$

The 3-D version of SC2 has the following form:

$$\min_{u,w\in[0,1]} J(u,w) = \int_\Omega |\nabla u|_g d\mathbf{x} + \int_\Omega rw \, d\mathbf{x} + \theta \int_\Omega P_d w \, d\mathbf{x} + \frac{1}{2\rho}\int_\Omega (u-w)^2 \, d\mathbf{x}. \tag{6.9}$$

Here, w is the new and dual variable, the right-most term enforces $w \approx u$ for sufficiently small $\rho > 0$, $r = (c_1 - z)^2 - (c_2 - z)^2$ and $|\nabla u|_g = g(|\nabla z|)|\nabla u|$. The similar edge detector function in [12, 150] is also used in the formulation to assist stopping the evolving curve on the edge of the object in an image. The area parameter θ is used to control the strength of the new addition distance fitting term. This means that if the parameter θ is too strong the final result will just be the polygon Q and if it is too weak the final result is potentially includes the nearby object.

The unique minimiser of J can be computed by minimising J with respect to u and w separately, iterating the process until convergence as done in [26, 20]. In alternating minimisation form, the new formulation of (6.9) is equivalent to solve the following functional

$$\min_u \; J_1(u,w) = \int_\Omega |\nabla u|_g d\mathbf{x} + \frac{1}{2\rho}\int_\Omega (u-w)^2 d\mathbf{x}, \tag{6.10}$$

$$\min_{w\in[0,1]} \; J_2(u,w) = \int_\Omega rw \, d\mathbf{x} + \theta \int_\Omega P_d w \, d\mathbf{x} + \frac{1}{2\rho}\int_\Omega (u-w)^2 \, d\mathbf{x}. \tag{6.11}$$

The explicit solution of (6.11) is given as

$$w = \min\{\max\{u(\mathbf{x}) - \rho r - \rho\theta P_d, 0\}, 1\}. \tag{6.12}$$

Now it only remains to discuss how to solve (6.10).

6.4 3-D optimisation based multilevel algorithm

This section presents our 3-D multilevel formulation to solve (6.10). For simplicity, we shall assume $n = 2^L$ for a given image z of size $N = n \times n \times n$. The standard coarsening defines $L+1$ levels: $s = 1$ (finest) $, 2, ..., L,\ L+1$ (coarsest) such that level s has $\tau_s \times \tau_s \times \tau_s$ "superpixels" with each "superpixels" having pixels $b_s \times b_s \times b_s$ where $\tau_s = n/2^{s-1}$ and $b_s = 2^{s-1}$. If $n \neq 2^L$, the multilevel method can still be developed with some coarse level superpixels of cube shapes and the rest of cuboid shapes.

We now consider our main model as expressed by (6.10)–(6.11). Minimisation of J is with respect to u in (6.10) and w in (6.11) respectively. The solution of (6.11) can be obtained analytically following equation 6.12. It remains to develop a multilevel algorithm to solve (6.10) in discretise-optimise scheme. The finite difference method is used to discretise (6.10) as done in related works [24, 33]. The discretised version of

(6.10) is given by

$$\begin{aligned}
&\min_u J_1(u) \equiv \min_u J_1^a(u_{1,1,1}, u_{2,1,1}, ..., u_{i-1,j,k}, u_{i,j,k}, u_{i+1,j,k}, ..., u_{n,n,n}) \\
&= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sum_{k=1}^{n-1} g_{i,j,k}\sqrt{(u_{i,j,k}-u_{i,j+1,k})^2 + (u_{i,j,k}-u_{i+1,j,k})^2 + (u_{i,j,k}-u_{i,j,k+1})^2 + \beta} \\
&+ \frac{1}{2\rho}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(u_{i,j,k}-w_{i,j,k})^2
\end{aligned} \tag{6.13}$$

where $g_{i,j,k} = g(x_i, y_j, z_j)$. Here u denotes a row vector.

As a prelude to multilevel methods, minimise (6.13) by a coordinate descent method (also known as relaxation algorithm) on the finest level 1:

$$\begin{cases}
\text{Given} & u^{(0)} = \left(u_{i,j,k}^{(0)}\right) \text{ with } m = 0, \\
\text{Solve} & u_{i,j,k}^{(m+1)} = \underset{u_{i,j,k}\in\mathbb{R}}{\arg\min} J_1^{loc}(u_{i,j}, c_1, c_2) \text{ for } i,j,k = 1,2,...n, \\
& \text{Repeat the above steps with } m = m+1 \text{ until stopped.}
\end{cases} \tag{6.14}$$

The equation (6.14) is obtained by expanding and simplifying the main model in (6.13) i.e.

$$\begin{aligned}
&J_1^{loc}(u_{i,j,k}) \\
&\equiv J_1^{loc}\left(u_{1,1,1}^{(m)}, u_{2,1,1}^{(m)}, ..., u_{i-1,j,k}^{(m)}, u_{i,j,k}, u_{i+1,j,k}^{(m)}, ..., u_{n,n,n}^{(m)}\right) - J_1^{(m)} \\
&= g_{i,j,k}\sqrt{\left(u_{i,j,k}-u_{i+1,j,k}^{(m)}\right)^2 + \left(u_{i,j,k}-u_{i,j+1,k}^{(m)}\right)^2 + \left(u_{i,j,k}-u_{i,j,k+1}^{(m)}\right)^2 + \beta} \\
&+ g_{i-1,j,k}\sqrt{\left(u_{i,j,k}-u_{i-1,j,k}^{(m)}\right)^2 + \left(u_{i-1,j,k}^{(m)}-u_{i-1,j+1,k}^{(m)}\right)^2 + \left(u_{i-1,j,k}^{(m)}-u_{i-1,j,k+1}^{(m)}\right)^2 + \beta} \\
&+ g_{i,j-1,k}\sqrt{\left(u_{i,j,k}-u_{i,j-1,k}^{(m)}\right)^2 + \left(u_{i,j-1,k}^{(m)}-u_{i+1,j-1,k}^{(m)}\right)^2 + \left(u_{i,j-1,k}^{(m)}-u_{i,j-1,k+1}^{(m)}\right)^2 + \beta} \\
&+ g_{i,j,k-1}\sqrt{\left(u_{i,j,k}-u_{i,j,k-1}^{(m)}\right)^2 + \left(u_{i,j,k-1}^{(m)}-u_{i+1,j,k-1}^{(m)}\right)^2 + \left(u_{i,j,k-1}^{(m)}-u_{i,j+1,k-1}^{(m)}\right)^2 + \beta} \\
&+ \tfrac{1}{2\rho}(u_{i,j,k}-w_{i,j,k})^2.
\end{aligned}$$

with Neumann's boundary condition applied where $J_1^{(m)}$ denotes the sum of all terms in J_1^a that do not involve $u_{i,j,k}$. Clearly one seems that this is a coordinate descent method.

The Newton method is used to solve the one-dimensional problem from (6.14) by

iterating $u^{(m)} \to u \to u^{(m+1)}$:

$$\begin{aligned}
&\frac{g_{i,j,k}\left(2u_{i,j,k}-u^{(m)}_{i+1,j,k}-u^{(m)}_{i,j+1,k}-u^{(m)}_{i,j,k+1}\right)}{\sqrt{\left(u_{i,j,k}-u^{(m)}_{i+1,j,k}\right)^2+\left(u_{i,j,k}-u^{(m)}_{i,j+1,k}\right)^2+\left(u_{i,j,k}-u^{(m)}_{i,j,k+1}\right)^2+\beta}}\\
&+\frac{g_{i-1,j,k}\left(u_{i,j,k}-u^{(m)}_{i-1,j,k}\right)}{\sqrt{\left(u_{i,j,k}-u^{(m)}_{i-1,j,k}\right)^2+\left(u_{i-1,j,k}-u^{(m)}_{i-1,j+1,k}\right)^2+\left(u_{i-1,j,k}-u^{(m)}_{i-1,j,k+1}\right)^2+\beta}}\\
&+\frac{g_{i,j-1,k}\left(u_{i,j,k}-u^{(m)}_{i,j-1,k}\right)}{\sqrt{\left(u_{i,j,k}-u^{(m)}_{i,j-1,k}\right)^2+\left(u_{i,j-1,k}-u^{(m)}_{i+1,j-1,k}\right)^2+\left(u_{i,j-1,k}-u^{(m)}_{i,j-1,k+1}\right)^2+\beta}}\\
&+\frac{g_{i,j,k-1}\left(u_{i,j,k}-u^{(m)}_{i,j,k-1}\right)}{\sqrt{\left(u_{i,j,k}-u^{(m)}_{i,j,k-1}\right)^2+\left(u_{i,j,k-1}-u^{(m)}_{i+1,j,k-1}\right)^2+\left(u_{i,j,k-1}-u^{(m)}_{i,j+1,k-1}\right)^2+\beta}}\\
&+\tfrac{1}{\rho}(u_{i,j,k}-w_{i,j,k})=0
\end{aligned}$$

giving rise to the form

$$u^{new}_{i,j,k} = u^{old}_{i,j,k} - T^{old}/B^{old} \tag{6.15}$$

where

$$\begin{aligned}
T^{old} = &\frac{g_{i,j,k}\left(2u^{old}_{i,j,k}-u^{(m)}_{i+1,j,k}-u^{(m)}_{i,j+1,k}-u^{(m)}_{i,j,k+1}\right)}{\sqrt{\left(u^{old}_{i,j,k}-u^{(m)}_{i+1,j,k}\right)^2+\left(u^{old}_{i,j,k}-u^{(m)}_{i,j+1,k}\right)^2+\left(u^{old}_{i,j,k}-u^{(m)}_{i,j,k+1}\right)^2+\beta}}\\
&+\frac{g_{i-1,j,k}\left(u^{old}_{i,j,k}-u^{(m)}_{i-1,j,k}\right)}{\sqrt{\left(u^{old}_{i,j,k}-u^{(m)}_{i-1,j,k}\right)^2+\left(u^{(m)}_{i-1,j,k}-u^{(m)}_{i-1,j+1,k}\right)^2+\left(u^{(m)}_{i-1,j,k}-u^{(m)}_{i-1,j,k+1}\right)^2+\beta}}\\
&+\frac{g_{i,j-1,k}\left(u^{old}_{i,j,k}-u^{(m)}_{i,j-1,k}\right)}{\sqrt{\left(u^{old}_{i,j,k}-u^{(m)}_{i,j-1,k}\right)^2+\left(u^{(m)}_{i,j-1,k}-u^{(m)}_{i+1,j-1,k}\right)^2+\left(u^{(m)}_{i,j-1,k}-u^{(m)}_{i,j-1,k+1}\right)^2+\beta}}\\
&+\frac{g_{i,j,k-1}\left(u^{old}_{i,j,k}-u^{(m)}_{i,j,k-1}\right)}{\sqrt{\left(u^{old}_{i,j,k}-u^{(m)}_{i,j,k-1}\right)^2+\left(u^{(m)}_{i,j,k-1}-u^{(m)}_{i+1,j,k-1}\right)^2+\left(u^{(m)}_{i,j,k-1}-u^{(m)}_{i,j+1,k-1}\right)^2+\beta}}\\
&+\tfrac{1}{\rho}(u_{i,j,k}-w_{i,j,k})
\end{aligned}$$

$$
\begin{aligned}
B^{old} = & \frac{2g_{i,j,k}}{\sqrt{\left(u^{old}_{i,j,k}-u^{(m)}_{i+1,j,k}\right)^2+\left(u^{old}_{i,j,k}-u^{(m)}_{i,j+1,k}\right)^2+\left(u^{old}_{i,j,k}-u^{(m)}_{i,j,k+1}\right)^2+\beta}} \\
& - \frac{g_{i,j,k}\left(2u^{old}_{i,j,k}-u^{(m)}_{i+1,j,k}-u^{(m)}_{i,j+1,k}-u^{(m)}_{i,j,k+1}\right)^2}{\sqrt{\left(\left(u^{old}_{i,j,k}-u^{(m)}_{i+1,j,k}\right)^2+\left(u^{old}_{i,j,k}-u^{(m)}_{i,j+1,k}\right)^2+\left(u^{old}_{i,j,k}-u^{(m)}_{i,j,k+1}\right)^2+\beta\right)^{\frac{3}{2}}} \\
& + \frac{g_{i-1,j,k}}{\sqrt{\left(u^{old}_{i,j,k}-u^{(m)}_{i-1,j,k}\right)^2+\left(u^{(m)}_{i-1,j,k}-u^{(m)}_{i-1,j+1,k}\right)^2+\left(u^{(m)}_{i-1,j,k}-u^{(m)}_{i-1,j,k+1}\right)^2+\beta}} \\
& - \frac{g_{i-1,j,k}\left(u^{old}_{i,j,k}-u^{(m)}_{i-1,j,k}\right)^2}{\sqrt{\left(\left(u^{old}_{i,j,k}-u^{(m)}_{i-1,j,k}\right)^2+\left(u^{(m)}_{i-1,j,k}-u^{(m)}_{i-1,j+1,k}\right)^2+\left(u^{(m)}_{i-1,j,k}-u^{(m)}_{i-1,j,k+1}\right)^2+\beta\right)^{\frac{3}{2}}} \\
& + \frac{g_{i,j-1,k}}{\sqrt{\left(u^{old}_{i,j,k}-u^{(m)}_{i,j-1,k}\right)^2+\left(u^{(m)}_{i,j-1,k}-u^{(m)}_{i+1,j-1,k}\right)^2+\left(u^{(m)}_{i,j-1,k}-u^{(m)}_{i,j-1,k+1}\right)^2+\beta}} \\
& - \frac{g_{i,j-1,k}\left(u^{old}_{i,j,k}-u^{(m)}_{i,j-1,k}\right)^2}{\sqrt{\left(\left(u^{old}_{i,j,k}-u^{(m)}_{i,j-1,k}\right)^2+\left(u^{(m)}_{i,j-1,k}-u^{(m)}_{i+1,j-1,k}\right)^2+\left(u^{(m)}_{i,j-1,k}-u^{(m)}_{i,j-1,k+1}\right)^2+\beta\right)^{\frac{3}{2}}} \\
& + \frac{g_{i,j,k-1}}{\sqrt{\left(u^{old}_{i,j,k}-u^{(m)}_{i,j,k-1}\right)^2+\left(u^{(m)}_{i,j,k-1}-u^{(m)}_{i+1,j,k-1}\right)^2+\left(u^{(m)}_{i,j,k-1}-u^{(m)}_{i,j+1,k-1}\right)^2+\beta}} \\
& - \frac{g_{i,j,k-1}\left(u^{old}_{i,j,k}-u^{(m)}_{i,j,k-1}\right)^2}{\sqrt{\left(\left(u^{old}_{i,j,k}-u^{(m)}_{i,j,k-1}\right)^2+\left(u^{(m)}_{i,j,k-1}-u^{(m)}_{i+1,j,k-1}\right)^2+\left(u^{(m)}_{i,j,k-1}-u^{(m)}_{i,j+1,k-1}\right)^2+\beta\right)^{\frac{3}{2}}} \\
& + \frac{1}{\rho}.
\end{aligned}
$$

To solve this coordinate descent method using multilevel method, we interpret solving (6.14) as looking for the best correction constant $\hat{c}$ at the current approximation $u^{(m)}_{i,j,k}$ on level 1 (the finest level) that minimises for c i.e.

$$
\min_{u_{i,j,k}\in\mathbb{R}} J_1^{loc}\left(u_{i,j,k}\right) = \min_{c\in\mathbb{R}} J_1^{loc}\left(u^{(m)}_{i,j,k}+c\right).
$$

Hence, we may rewrite (6.14) in an equivalent form:

$$
\begin{cases}
\text{Given} & u^{(0)} = \left(u^{(0)}_{i,j,k}\right) \quad \text{with } m=0, \\
\text{Solve} & \hat{c} = \arg\min_{c\in\mathbb{R}} J_1^{loc}\left(u^{(m)}_{i,j,k}+c, c_1, c_2\right) \text{ for } i,j=1,2,...n, \\
\text{Update} & u^{(m+1)}_{i,j,k} = u^{(m)}_{i,j,k}+\hat{c}, \\
 & \text{Repeat the above steps with } m=m+1 \text{ until stopped.}
\end{cases}
\tag{6.16}
$$

The remaining task is to derive the simplified formulation for each of the subproblems associated with these blocks on level s. Figure 6.1 illustrates multilevel method for level 3 of image size $16\times16\times16$. (a) shows one of $\tau_3^3 = 4^3$ superpixel in level 3. Each superpixel contains $b_3^3 = 4^3$ pixels. (b) represents the top surface of (a). Using equation (6.13), the interaction of a pixel with neighbouring pixel (red $\bullet$) is illustrate in (c). (d) shows the interaction of pixels in (b) based on equation (6.13) and (c).

We set the following: $b = 2^{s-1}$, $k_1 = (i-1)\,b+1$, $k_2 = ib$, $\ell_1 = (j-1)\,b+1$, $\ell_2 = jb$, $m_1 = (k-1)\,b+1$, $m_2 = kb$, and $c = (c_{i,j,k})$. Denoted the current solution $\tilde{u}$ then, a

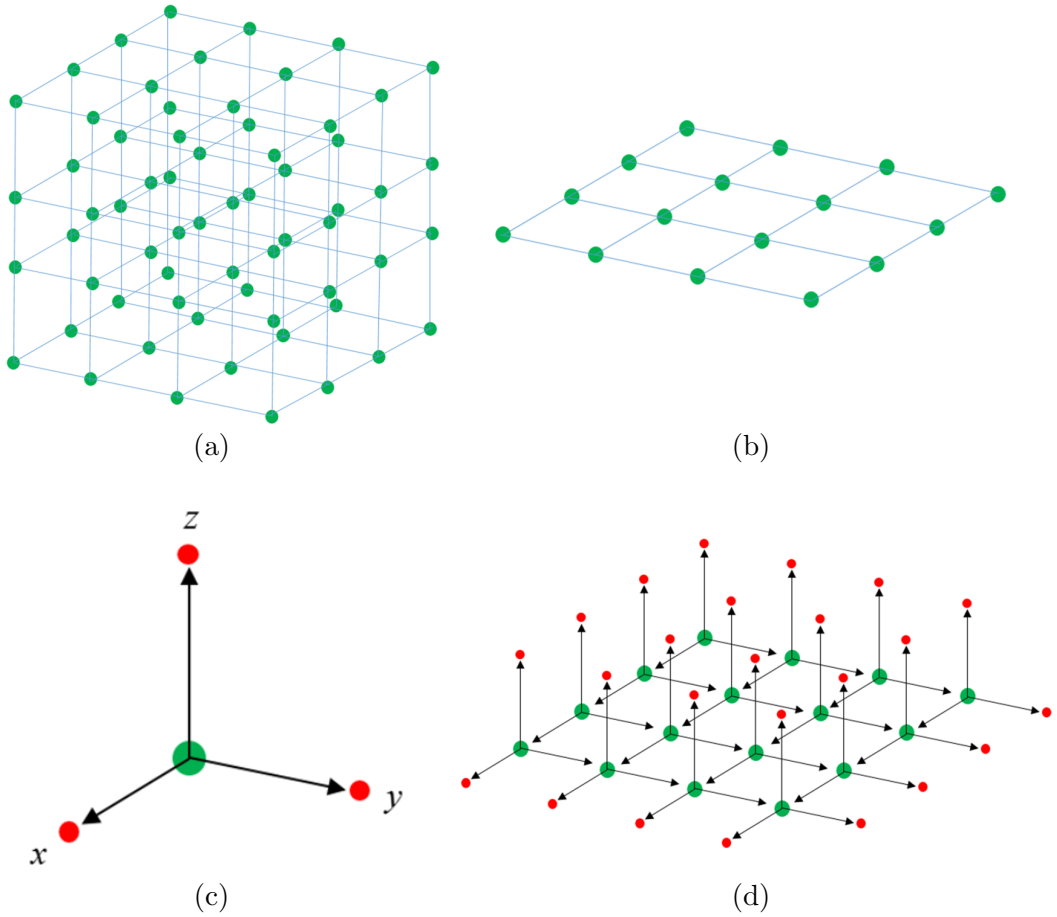


Figure 6.1: Illustration of level 3 for image size $16 \times 16 \times 16$. (a) shows one of $\tau_3^3 = 4^3$ superpixel in level 3. Each superpixel contains $b_3^3 = 4^3$ pixels. (b) represents the top surface of (a). Using equation (6.13), the interaction of a pixel with neighbouring pixel (red $\bullet$) is illustrate in (c). (d) shows the interaction of pixels in (b) based on equation (6.13) and (c).

general computational stencil involving c on level s can be illustrated as follows

(6.17)

Then, minimising for c, the problem 6.16 is equivalent to minimise the following

$$
\begin{aligned}
F_{3DSC2}(c_{i,j,k}) = &\sum_{k=k_1}^{k_2-1} g_{k,\ell_2,m_1}\sqrt{T_1+\beta} + \sum_{k=\ell_1}^{\ell_2-1} g_{k_2,\ell,m_1}\sqrt{T_2+\beta} + g_{k_2,\ell_2,m_1}\sqrt{T_3+\beta} \\
&+ \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_1,\ell,m_2}\sqrt{T_4+\beta} + \sum_{k=k_1}^{k_2-1} g_{k,\ell_2,m_2}\sqrt{T_5+\beta} + g_{k_2,\ell_2,m_2}\sqrt{T_6+\beta} \\
&+ \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell,m_2}\sqrt{T_7+\beta} + \sum_{k=k_1}^{k_2-1} g_{k,\ell_1,m_2}\sqrt{T_8+\beta} + \sum_{m=m_1+1}^{m_2-1} g_{k_1,\ell_2,m}\sqrt{T_9+\beta} \\
&+ \sum_{m=m_1+1}^{m_2-1} g_{k_2,\ell_2,m}\sqrt{T_{10}+\beta} + \sum_{m=m_1+1}^{m_2-1} g_{k_2,\ell_1,m}\sqrt{T_{11}+\beta} \\
&+ \sum_{k=k_1+1}^{k_2-1}\sum_{\ell=\ell_1+1}^{\ell_2-1} g_{k,\ell,m_2}\sqrt{T_{12}+\beta} + \sum_{\ell=\ell_1+1}^{\ell_2-1}\sum_{m=m_1+1}^{m_2-1} g_{k_2,\ell,m}\sqrt{T_{13}+\beta} \\
&+ \sum_{k=k_1+1}^{k_2-1}\sum_{m=m_1+1}^{m_2-1} g_{k,\ell_2,m}\sqrt{T_{14}+\beta} + \sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2} g_{k,\ell,m_1-1}\sqrt{T_{15}+\beta} \\
&+ \sum_{\ell=\ell_1}^{\ell_2}\sum_{m=m_1}^{m_2} g_{k_1-1,\ell,m}\sqrt{T_{16}+\beta} + \sum_{k=k_1}^{k_2}\sum_{m=m_1}^{m_2} g_{k,\ell_1-1,m}\sqrt{T_{17}+\beta} \\
&+ \tfrac{1}{2\rho}\sum_{k=k_1}^{k_2}\sum_{\ell=\ell 1}^{\ell_2}\sum_{m=m_1}^{m_2} (u_{k,\ell,m} + c_{i,j,k} - w_{k,\ell,m})^2.
\end{aligned}
\tag{6.18}
$$

where

$$
\begin{aligned}
T_1 &= (\tilde{u}_{k,\ell_2,m_1} + c_{i,j,k} - \tilde{u}_{k,\ell_2+1,m_1})^2 + (\tilde{u}_{k,\ell_2,m_1} - \tilde{u}_{k+1,\ell_2,m_1})^2 \\
&\quad +(\tilde{u}_{k,\ell_2,m_1} - \tilde{u}_{k,\ell_2,m_1+1})^2 \\
T_2 &= (\tilde{u}_{k_2,\ell,m_1} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell,m_1})^2 + (\tilde{u}_{k_2,\ell,m_1} - \tilde{u}_{k_2,\ell+1,m_1})^2 \\
&\quad +(\tilde{u}_{k_2,\ell,m_1} - \tilde{u}_{k_2,\ell,m_1+1})^2 \\
T_3 &= (\tilde{u}_{k_2,\ell_2,m_1} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell_2,m_1})^2 + (\tilde{u}_{k_2,\ell_2,m_1} + c_{i,j,k} - \tilde{u}_{k_2,\ell_2+1,m_1})^2 \\
&\quad +(\tilde{u}_{k_2,\ell_2,m_1} - \tilde{u}_{k_2,\ell_2,m_1+1})^2 \\
T_4 &= (\tilde{u}_{k_1,\ell,m_2} + c_{i,j,k} - \tilde{u}_{k_1,\ell,m_2+1})^2 + (\tilde{u}_{k_1,\ell,m_2} - \tilde{u}_{k_1+1,\ell,m_2})^2 \\
&\quad +(\tilde{u}_{k_1,\ell,m_2} - \tilde{u}_{k_1,\ell+1,m_2})^2 \\
T_5 &= (\tilde{u}_{k,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k,\ell_2+1,m_2})^2 + (\tilde{u}_{k,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k,\ell_2,m_2+1})^2 \\
&\quad +(\tilde{u}_{k,\ell_2,m_2} - \tilde{u}_{k+1,\ell_2,m_2})^2 \\
T_6 &= (\tilde{u}_{k_2,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell_2,m_2})^2 + (\tilde{u}_{k_2,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k_2,\ell_2+1,m_2})^2 \\
&\quad +(\tilde{u}_{k_2,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k_2,\ell_2,m_2+1})^2 \\
T_7 &= (\tilde{u}_{k_2,\ell,m_2} + c_{i,j,k} - \tilde{u}_{k_2,\ell,m_2+1})^2 + (\tilde{u}_{k_2,\ell,m_2} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell,m_2})^2 \\
&\quad +(\tilde{u}_{k_2,\ell,m_2} - \tilde{u}_{k_2,\ell+1,m_2})^2 \\
T_8 &= (\tilde{u}_{k,\ell_1,m_2} + c_{i,j,k} - \tilde{u}_{k,\ell_1,m_2+1})^2 + (\tilde{u}_{k,\ell_1,m_2} - \tilde{u}_{k,\ell_1+1,m_2})^2 \\
&\quad +(\tilde{u}_{k,\ell_1,m_2} - \tilde{u}_{k+1,\ell_1,m_2})^2 \\
T_9 &= (\tilde{u}_{k_1,\ell_2,m} + c_{i,j,k} - \tilde{u}_{k_1,\ell_2+1,m})^2 + (\tilde{u}_{k_1,\ell_2,m} - \tilde{u}_{k_1,\ell_2,m+1})^2 \\
&\quad +(\tilde{u}_{k_1,\ell_2,m} - \tilde{u}_{k_1+1,\ell_2,m})^2 \\
T_{10} &= (\tilde{u}_{k_2,\ell_2,m} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell_2,m})^2 + (\tilde{u}_{k_2,\ell_2,m} + c_{i,j,k} - \tilde{u}_{k_2,\ell_2+1,m})^2 \\
&\quad +(\tilde{u}_{k_2,\ell_2,m} - \tilde{u}_{k_2,\ell_2,m+1})^2 \\
T_{11} &= (\tilde{u}_{k_2,\ell_1,m} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell_1,m})^2 + (\tilde{u}_{k_2,\ell_1,m} - \tilde{u}_{k_2,\ell_1+1,m})^2 \\
&\quad +(\tilde{u}_{k_2,\ell_1,m} - \tilde{u}_{k_2,\ell_1,m+1})^2 \\
T_{12} &= (\tilde{u}_{k,\ell,m_2} + c_{i,j,k} - \tilde{u}_{k,\ell,m_2+1})^2 + (\tilde{u}_{k,\ell,m_2} - \tilde{u}_{k+1,\ell,m_2})^2 \\
&\quad +(\tilde{u}_{k,\ell,m_2} - \tilde{u}_{k,\ell+1,m_2})^2 \\
T_{13} &= (\tilde{u}_{k_2,\ell,m} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell,m})^2 + (\tilde{u}_{k_2,\ell,m} - \tilde{u}_{k_2,\ell+1,m})^2 \\
&\quad +(\tilde{u}_{k_2,\ell,m} - \tilde{u}_{k_2,\ell,m+1})^2 \\
T_{14} &= (\tilde{u}_{k,\ell_2,m} + c_{i,j,k} - \tilde{u}_{k,\ell_2+1,m})^2 + (\tilde{u}_{k,\ell_2,m} - \tilde{u}_{k+1,\ell_2,m})^2 \\
&\quad +(\tilde{u}_{k,\ell_2,m} - \tilde{u}_{k,\ell_2,m+1})^2 \\
T_{15} &= (\tilde{u}_{k,\ell,m_1} + c_{i,j,k} - \tilde{u}_{k,\ell,m_1-1})^2 + (\tilde{u}_{k,\ell,m_1-1} - \tilde{u}_{k+1,\ell,m_1-1})^2 \\
&\quad +(\tilde{u}_{k,\ell,m_1-1} - \tilde{u}_{k,\ell+1,m_1-1})^2 \\
T_{16} &= (\tilde{u}_{k_1,\ell,m} + c_{i,j,k} - \tilde{u}_{k_1-1,\ell,m})^2 + (\tilde{u}_{k_1-1,\ell,m} - \tilde{u}_{k_1-1,\ell+1,m})^2 \\
&\quad +(\tilde{u}_{k_1-1,\ell,m} - \tilde{u}_{k_1-1,\ell,m+1})^2 \\
T_{17} &= (\tilde{u}_{k,\ell_1,m} + c_{i,j,k} - \tilde{u}_{k,\ell_1-1+1,m})^2 + (\tilde{u}_{k,\ell_1-1,m} - \tilde{u}_{k,\ell_1-1,m+1})^2 \\
&\quad +(\tilde{u}_{k,\ell_1-1,m} - \tilde{u}_{k+1,\ell_1-1,m})^2
\end{aligned}
$$

On the coarsest level $(L+1)$, a **single** constant update for the current $\tilde{u}$ is given as

$$
\min_c \{ F_{3DSC2}(\tilde{u} + c) = \frac{1}{2\rho} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (u_{i,j,k} + c - w_{i,j,k})^2 \} \tag{6.19}
$$

which has a simple and explicit solution.

The solutions of the above local minimisation problems, solved using a Newton

method as in (6.15) or a fixed point method for t iterations (inner iteration), defines the update solution $u = u + Q_s c$ where Q_s is the interpolation operator distributing $c_{i,j,k}$ to the corresponding $b_s \times b_s \times b_s$ block on level s as illustrated in (6.17). Then we obtain a multilevel method if we cycle through all levels and all blocks on each level until $\max\left(\frac{\|\tilde{u}-u\|_2}{\|\tilde{u}\|_2}, \frac{\|\tilde{w}-w\|_2}{\|\tilde{w}\|_2}\right) < tol$ or the maximum number of cycle, $maxit$ is reached.

Finally our proposed multilevel method for 3-D SC2 is summarized in Algorithm 7. We will use the term **3DSC2** to refer this 3-D multilevel Algorithm 7.

Algorithm 7 3DSC2 – Algorithm to solve the new primal-dual model

---

Given image z, an initial guess u , the stop tolerance (tol), and maximum multilevel cycle ($maxit$) with $L+1$ levels. Set $w = u$,

1) *Solve* (6.10) *to update u using the following steps:*

   i). *Set $\tilde{u} = u$.*

   ii). *Smooth for t iteration the approximation on the finest level 1 that is solve* (6.14) *for $i, j, k = 1, 2, ...n$*

   iii). *Iterate for t times on each coarse level $s = 2, 3, ...L, L+1$:*
   > If $s \leq L$, compute the minimiser c of (6.18)
   > Solve (6.19) *on the coarsest level $s = L+1$*
   > Add the correction $u = u + Q_s c$ where Q_s is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_s \times b_s \times b_s$ block on level s as illustrated in (6.17).

2) *Solve* (6.11) *to update w:*

   i). *Set $\tilde{w} = w$.*

   ii). *Compute w using the formula* (6.12).

3) *Check for convergence using the above criteria. If not satisfied, return to Step 1. Otherwise exit with solution $u = \tilde{u}$ and $w = \tilde{w}$*

We recommended to start updating our multilevel algorithm from the fine level to the coarse level in order to get fast convergence. In a separate experiment we found that if we adjust the coarse structure before the fine level, the convergence is slower. In addition, the value of inner iteration $t = 1$ is recommended to update the algorithm in a fast manner.

6.5 A new localised model

Solution of (6.10) and (6.11) can be expensive in the whole 3-D domain Ω. Here, we introduce our approach for reducing the computation time for the type of problem. Given a level-set function ϕ, local functions b_1 and b_2 defined by

$$b_1(\phi(\mathbf{x}), \gamma) = 1 - H\left(\phi(\mathbf{x}) - \gamma\right)$$

$$b_2(\phi(\mathbf{x}), \gamma) = H\left(\phi(\mathbf{x}) + \gamma\right)$$

characterizes the domain of narrow band region $\Omega_\gamma = \Omega_1(\gamma) \cup \Gamma \cup \Omega_2(\gamma)$ around the 2-D surface Γ where we assume ϕ is positive inside the surface Γ and negative outside it. Here, $\Omega_1(\gamma)$ and $\Omega_2(\gamma)$ represents the γ-band region inside and outside of the surface Γ respectively. Inside the domain Ω_γ , the value of $b_1 = b_2 = 1$ while outside Ω_γ, $b_1 = b_2 = 0$. Similarly, $b_1 H(\phi) = 1$ inside $\Omega_1(\gamma)$ and 0 outside and we have $b_1(1 - H(\phi)) = 1$ inside $\Omega_2(\gamma)$ and 0 outside.

We first modified the fitting term of our 3-D SC2 model as follows

$$\int_{\Omega_1} (z - c_1)^2 \, d\mathbf{x} \to \int_{\Omega} (z - c_1)^2 b_1(\phi, \gamma) H(\phi) \, d\mathbf{x}$$
$$\int_{\Omega_2} (z - c_1)^2 \, d\mathbf{x} \to \int_{\Omega} (z - c_2)^2 b_2(\phi, \gamma)(1 - H(\phi)) \, d\mathbf{x}$$

This notation is almost similar to [152, 150] except for each integral in level set above, our approach use only two heaviside functions, $H(\phi)$ while there are three $H(\phi)$ used in [152, 150]. Consequently, our formulation is less complex than [152, 150]. Next, the term $H(\phi) \in \{0, 1\}$ is relaxed to $u \in [0, 1]$ as used in [34]. As a common practice in image segmentation, the final solution $u(\mathbf{x})$ defining the targeted object by $\sum = \{(x) : u(x) \geq \eta\}$ and usually $\eta = 0.5$. With the above connection, we set $\phi = u - \eta$. Consequently, the local functions b_1 and b_2 are defined by

$$b_1(u(\mathbf{x}), \gamma) = 1 - H\left(u(\mathbf{x}) - \eta - \gamma\right)$$

$$b_2(u(\mathbf{x}), \gamma) = H\left(u(\mathbf{x}) - \eta + \gamma\right)$$

Using the above information, the new localised version of 3D-SC2 model is defined below

$$\begin{aligned} \min_{u,w \in [01]} J_L(u, w) = &\int_\Omega |\nabla u|_g d\mathbf{x} + \int_\Omega (z - c_1)^2 w b_1 \, d\mathbf{x} + \theta \int_\Omega P_d w d\mathbf{x} \\ &+ \int_\Omega (z - c_2)^2 (1 - w) b_2 \, d\mathbf{x} + \frac{1}{2\rho} \int_\Omega (u - w)^2 d\mathbf{x} \end{aligned} \tag{6.20}$$

Minimisation with respect to u and w yields the following functional

$$\begin{aligned} \min_u J_{L1}(u, w) = &\int_\Omega |\nabla u|_g d\mathbf{x} + \int_\Omega (z - c_1)^2 w b_1 \, d\mathbf{x} + \frac{1}{2\rho} \int_\Omega (u - w)^2 d\mathbf{x} \\ &+ \int_\Omega (z - c_2)^2 (1 - w) b_2 \, d\mathbf{x} \end{aligned} \tag{6.21}$$

$$\begin{aligned} \min_{w \in [01]} J_{L2}(u, w) = &\int_\Omega (z - c_1)^2 w b_1 \, d\mathbf{x} + \int_\Omega (z - c_2)^2 (1 - w) b_2 \, d\mathbf{x} \\ &+ \theta \int_\Omega P_d w d\mathbf{x} + \frac{1}{2\rho} \int_\Omega (u - w)^2 d\mathbf{x} \end{aligned} \tag{6.22}$$

For (6.22), the explicit solution is given as

$$w = \min\left\{\max\left\{u - \left((z-c_1)^2 b_1 - (z-c_2)^2 b_2\right)\rho - \rho\theta P_d, 0\right\}, 1\right\} \tag{6.23}$$

In order to apply multilevel algorithm to solve (6.21), we introduce (after discretisation) the notation for the set falling into the γ-band where $b_1 = b_2 = 1$:

$$D(u) = \{(i,j,k)\,|-\gamma \leq u_{i,j,k} \leq \gamma\}$$

All the steps to solve (6.21) using multilevel algorithm are identical except $u = u + c_{i,j,k}$ only needs an updated if the set $[k_1, k_2] \times [\ell_1, \ell_2] \times [m_1, m_2] \cap D(u)$ is non-empty. We will call **3DSC3** to refer the multilevel algorithm to solve localised version of 3D-SC2 model.

6.6 New variant of the multilevel algorithms 3DSC2 and 3DSC3

Our above proposed method for an optimisation problem defines a sequence of search directions based in a multilevel setting . In this section,we modify it so that the new algorithm has a formal decaying property.

Denote the functional in (6.13) by $g(u) :\ \mathbb{R}^{n^3} \to \mathbb{R}$ and represent each subproblem by

$$c^* = \arg\min_{c\in\mathbb{R}} g(u^\ell + cp^\ell), \quad u^{\ell+1} = u^\ell + c^* p^\ell, \quad p^\ell = \tilde{\mathbf{e}}^{\ell(\mathrm{mod}\ K)+1}, \quad \ell = 0, 1, 2, \ldots$$

where $\tilde{\mathbf{e}}$ and K will be defined below, $\mathrm{mod}(\cdot)$ denotes modulo operator. Noting that $b_s = 2^{s-1}$ and $\tau_s = n/b_s$.

Now we investigate the search direction $\{\tilde{\mathbf{e}}\}$;

$$
\begin{aligned}
\text{level } 1 \quad &: \quad 1\times 1 \text{ block}'\text{s index } (i,j,k), \quad 1\le i,j,k\le n\\
&\quad \tilde{\mathbf{e}}^f = \mathbf{e}_f, \quad f=1,2,...n^3\\
\text{level } 2 \quad &: \quad 2^1\times 2^1 \text{ block}'\text{s indices } (k_1:k_2,\ell_1:\ell_2,m_1:m_2), \quad 1\le i,j,k\le \tau_2\\
&\quad \tilde{\mathbf{e}}^{K_0+(k-1)\tau_2^2+(j-1)\tau_2+i} = \sum_{i_1=k_1}^{k_2}\sum_{j_1=\ell_1}^{\ell_2}\sum_{q_1=m_1}^{m_2}\mathbf{e}_{n^2(q_1-1)+n(j_1-1)+i_1}\\
&\quad k_1=2^1(i-1)+1,\ k_2=2^1 i;\ \ell_1=2^1(j-1)+1,\ \ell_2=2^1 j;\\
&\quad m_1=2^1(k-1)+1,\ m_2=2^1 k\\
\text{level } 3 \quad &: \quad 2^2\times 2^2 \text{ block}'\text{s indices } (k_1:k_2,\ell_1:\ell_2,m_1:m_2), \quad 1\le i,j,k\le \tau_3\\
&\quad \tilde{\mathbf{e}}^{K_0+K_1+(k-1)\tau_3^2+(j-1)\tau_3+i} = \sum_{i_1=k_1}^{k_2}\sum_{j_1=\ell_1}^{\ell_2}\sum_{q_1=m_1}^{m_2}\mathbf{e}_{n^2(q_1-1)+n(j_1-1)+i_1}\\
&\quad k_1=2^2(i-1)+1,\ k_2=2^2 i;\ \ell_1=2^2(j-1)+1,\ \ell_2=2^2 j;\\
&\quad m_1=2^2(k-1)+1,\ m_2=2^2 k\\
&\quad \vdots\\
\text{level } L+1 \quad &: \quad 2^L\times 2^L \text{ block}'\text{s indices } (1:n,1:n,1:n)\\
&\quad \tilde{\mathbf{e}}^K = \sum_{i_1=1}^{n}\sum_{j_1=1}^{n}\sum_{q_1=1}^{n}\mathbf{e}_{n^2(q_1-1)+n(j_1-1)+i_1} = \sum_{V=1}^{n^3}\mathbf{e}_V
\end{aligned}
$$

where $\mathbf{e}_\iota \in {}^{n^3}$ is the ι-th unit (coordinate) vector,

$$
\begin{aligned}
K = \sum_{Y=0}^{L} K_Y &= \sum_{Y=0}^{L}\tau_Y^2\\
&= \sum_{Y=0}^{L}\frac{n^3}{8^Y} = \frac{8n^3-1}{7}
\end{aligned}
$$

Here K represents the total number of search direction across all levels $1,2,...L,L+1$ for this unconstrained type optimisation problem. As shown above, the sequence $\{p^\ell\}$ is clearly essentially periodic (finitely many) and free-steering (spanning $\mathbb{R}^{n^3}$) [105].

Recall that a sequence $\{u^\ell\}$ is strongly downward (decaying) with respect to $g(u)$ i.e.

$$
g(u^\ell)\ge g(v^\ell)\ge g(u^{\ell+1}), \quad v^\ell=(1-t)u^\ell+tu^{\ell+1}\in D_0, \ \ \forall\ t\in[0,1]. \tag{6.24}
$$

This property is much stronger than the usual decaying property $g(u^\ell)\ge g(u^{\ell+1})$ which is automatically satisfied by our Algorithm **3DSC2 and 3DSC3**.

By [105, Thm 14.2.7], to ensure the minimising sequence $\{u_\ell\}$ to be strongly downward, we modify the subproblem $\min J_1^{loc}(u^\ell+cp^\ell)$ to the following

$$
u^{\ell+1}=u^\ell+c^*q^\ell, \qquad c^*=\arg\min\{c\ge 0 \mid \nabla J^T q^\ell=0\}, \qquad \ell\ge 0 \tag{6.25}
$$

where the ℓ-th search direction is modified to

$$q^\ell = \begin{cases} p^\ell, & \text{if } \nabla J^T p^\ell \leq 0, \\ -p^\ell, & \text{if } \nabla J^T p^\ell > 0. \end{cases}$$

Here the equation $\nabla J^T q^\ell = 0$ for c and the local minimising subproblem (6.16) that is $\min_c J_1^{loc}(\hat{u}_{i,j} + c)$ are equivalent. Now the new modification is to enforce $c \geq 0$ and the sequence $\{q^\ell\}$ is still essentially periodic.

We shall call the modified algorithm **3DSC2** and **3DSC3** as **3DSC2M** and **3DSC3M** respectively.

6.7 Convergence and complexity analysis

The convergence of the algorithms **3DSC2-3DSC3** for

$$\min_{u \in \mathbb{R}} g(u)$$

is challenging to prove unless a stronger assumption of uniform convexity for the minimisation of functional g is made. However there is another approach to prove the convergence of **3DSC2M-3DSC3M** for solving problem (6.13) without such an assumption. We first assume the functional $g = g(u)$ is hemivariate i.e. $g(u+t(v-u)) = g(u)$ for t in $[0, 1]$ and $u \neq v$ (for theoretical purpose).

There are five sufficient conditions that must be met in order to prove the convergence of **3DSC2M-3DSC3M**,

i) $g(u)$ is continuously differentiable in $D_0 = [0, 1]^{n^3} \subset \mathbb{R}^{n^3}$;

ii) the sequence $\{q^\ell\}$ is uniformly linearly independent;

iii) the sequence $\{u^\ell\}$ is strongly downward (decaying) with respect to $g(u)$;

iv) $\lim_{\ell \to \infty} g'(u^\ell) q^\ell / \|q^\ell\| = 0$,

v) the set $S = \{u \in D_0 \mid g'(u) = 0\}$ is non-empty.

Here $g'(u) = (\nabla g(u))^T$. Then we have the convergence of $\{u^\ell\}$ to a critical point u^* [105, Thm 14.1.4]

$$\lim_{\ell \to \infty} \inf_{u \in S} \|u^\ell - u^*\| = 0.$$

The condition i) is met since we set $\beta \neq 0$ and condition ii) also holds since 'essentially periodic' implies 'uniformly linearly independent' [105, §14.6.3]. We make an assumption of existence of stationary points for $g(u)$ to verify condition v). Next, we now focus on verifying condition iii)-iv). From [105, Thm 14.2.7], the construction of $\{u^\ell\}$ via (6.25) ensures that the sequence $\{u^\ell\}$ is strongly downward and further $\lim_{\ell \to \infty} g'(u^\ell) q^\ell / \|q^\ell\| = 0$. Hence conditions iii)-iv) are satisfied.

Note condition iii) and the assumption of $g(u)$ being hemivariate imply $\lim_{\ell\to\infty}\|u^{\ell+1}-u^{\ell}\|=0$ from [105, Thm 14.1.3]. Further condition iv) and the fact $\lim_{\ell\to\infty}\|u^{\ell+1}-u^{\ell}\|=0$ lead to the result $\lim_{\ell\to\infty} g'(u^{\ell})=\mathbf{0}$.

Finally by [105, Thm 14.1.4], the condition $\lim_{\ell\to\infty} g'(u^{\ell})=\mathbf{0}$ implies $\lim_{\ell\to\infty}\inf_{u\in S}\|u^{\ell}-u^{*}\|=0$. Hence the convergence is proved.

Next, we will give the complexity analysis of our **3DSC2**, **3DSC3**, **3DSC2M** and **3DSC3M**. Let $N=n^3$ be the total number of pixels (unknowns). First, we compute the number of floating point operations (flops) for **3DSC2** for level s as follows:

Quantities	Flop counts for **3DSC2**
ρ term	$2N$
w term	$6N$
s smoothing steps	$125b_s\tau_s^3 r$

Then, the flop counts for all level is $W_{3DSC2}=6N+\sum_{s=1}^{L+1}\left(2N+125b_s\tau_s^3 r\right)$ where $s=1$ (finest) and $s=L+1$ (coarsest). Noting $b_s=2^{s-1},\tau_s=n/b_s,N=n^3$, we compute the upper bound for **3DSC2** as follows:

$$W_{3DSC2}=6N+2(L+1)N+\sum_{s=1}^{L+1}\left(\frac{125Nr}{b_s}\right)=6N+2(L+1)N+(125r)N\sum_{s=0}^{L}\left(\frac{1}{2^s}\right)$$
$$<2N\log n+8N+250Nr\approx O\left(N\log N\right)$$

The complexity of the new **3DSC3** is directly linked to the length of the segmented objects at each iteration; at the discrete level, this length is usually $O\left(\sqrt[3]{N}\right)$. Similarly the upper bound for **3DSC3** is computed as follows:

$$W_{3DSC3}=6\sqrt[3]{N}+2(L+1)\sqrt[3]{N}+\sum_{s=1}^{L+1}\left(\frac{135r\sqrt[3]{N}}{b_s}\right)=6\sqrt[3]{N}+2(L+1)\sqrt[3]{N}$$
$$+(135r)\sqrt[3]{N}\sum_{s=0}^{L}\left(\tfrac{1}{2^s}\right)<2\sqrt[3]{N}\log n+8\sqrt[3]{N}+270r\sqrt[3]{N}\approx O\left(\sqrt[3]{N}\log N\right)$$

The approximate cost of an extra operation $\nabla J^T q^{\ell}$ in **3DSC2M** and **3DSC3M** is $2N$ that results to the total flop counts for **3DSC2M** as $W_{3DSC2M}=6N+\sum_{s=1}^{L+1}\left(4N+125b_s\tau_s^3 r\right)$. This gives the upper bound for **3DSC2M** as

$$W_{3DSC2M}=6N+4(L+1)N+\sum_{s=1}^{L+1}\left(\frac{125Nr}{b_s}\right)=6N+4(L+1)N+$$
$$(125r)N\sum_{s=0}^{L}\left(\tfrac{1}{2^s}\right)<4N\log n+10N+250Nr\approx O\left(N\log N\right).$$

Finally, for **3DSC3M**, the upper bound for **3DSC3M** is computed as follows:

$$W_{3DSC3M} = 6\sqrt[3]{N} + 4(L+1)\sqrt[3]{N} + \sum_{s=1}^{L+1}\left(\frac{135r\sqrt[3]{N}}{b_s}\right) = 6\sqrt[3]{N} + 4(L+1)\sqrt[3]{N}$$
$$+(135r)\sqrt[3]{N}\sum_{s=0}^{L}\left(\tfrac{1}{2^s}\right) < 2\sqrt[3]{N}\log n + 10\sqrt[3]{N} + 270r\sqrt[3]{N} \approx O\left(\sqrt[3]{N}\log N\right)$$

One can observe that **3DSC2**, **3DSC3**, **3DSC2M** and **3DSC3M** are of the optimal complexity expected of a multilevel method and $W_{3DSC3} < W_{3DSC3M} < W_{3DSC2} < W_{3DSC2M}$.

6.8 Numerical experiment

This section will demonstrate the performance of the developed multilevel methods through several experiments. The algorithms to be compared are:

Name	**Algorithm**		**Description**
3DSC2	New	:	The 3-D multilevel Algorithm 7 for the 3-D version of SC2 model
3DSC3	New	:	The multilevel algorithm for localised version of 3DSC2.
3DSC2M	New	:	The modified multilevel algorithm for 3DSC2
3DSC3M	New	:	The modified multilevel algorithm for 3DSC3.
3DCHB	Old	:	The 3-D Chambolle's Algorithm for the 3-D version of SC2 model, see Appendix A.
3DZCG	Old	:	The 3-D selective segmentation model by [150].

There are four sets of tests carried out. In the **first set**, we will compare the performance of algorithm 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M in segmenting 3-D artificial geometrical image with some noise added in different resolutions. We record their segmentation performances in terms of CPU time (in seconds) and quality. The segmentation quality is measured based on the Jaccard similarity coefficient (JSC):

$$JSC = \frac{|S_n \cap S_*|}{|S_n \cup S_*|}$$

where S_n is the set of the segmented domain u and S_* is the true set of u (which is only easy to obtain for simple images). The similarity functions return values in the range $[0, 1]$. The value 1 indicates perfect segmentation quality while the value 0 indicates poor quality.

Our **second test** are conducted on segmenting harder and more challenging 3-D medical data using 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M. We will choose the best multilevel algorithm based on the tests in **first set** and **second set**. The choosen multilevel algorithm will be compared with 3DCHB (see Appendix A) and 3DZCG [150] in the **third** and **fourth set** respectively.

Figure 6.2(a) shows the image used in test set 1, 6.2(b) and (c) are for test set 2 while 6.2(d) is used for test 3. The images in Figure 6.7(a) and (b) are used in test

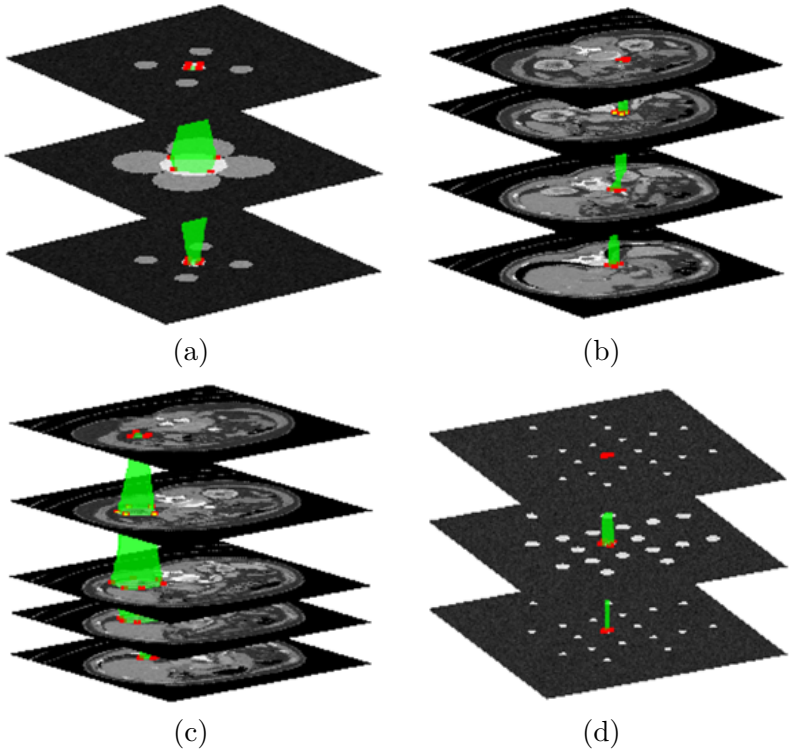


Figure 6.2: (a) is an image used in test set 1, (b) and (c) for test set 2 and (d) for test set 3. Markers set are in red and the green polyhedral surface constructed based on the position of markers set are the initial solution.

set 4. The images in Figure 6.2 are of interest because they involve both real medical and synthetic images. The markers set are in red and the green polyhedral surface constructed based on the position of markers set are the initial solution. In case for an object that is difficult to be distinguished from the background, the idea of using the edge detector before segmentation process can be used to determine the suitable position of markers and initialisation. The targeted objects used in 6.2(b) and (c) have small separation gap, however the objects are still can be distinguished from the background. The location of markers and initialisations for the objects in Figure 6.2(a) and (d) are easy to determine as they are synthetic images. Thus, the idea of applying edge detector before segmentation process is not implemented in this chapter. In addition, we have added the edge detection function in our formulation which help to stop the evolving curve on the edge of the object. All algorithms are implemented in MATLAB R2017a on a computer with Intel Core i7 processor, CPU 3.60GHz, 16 GB RAM CPU.

In the following experiment, we used parameter $\beta = 10^{-4}$, ρ in between 7×10^{-4} and 10^{-3}, γ in between 1 and 10, ε in between $1/n^2$ and 0.01, q in between 1 and 10 and $maxit = 8000$ to get a successful segmentation result. Tuning the parameter θ depends on the targeted object. If the object is too close to a nearby boundary then θ should be large. Segmenting a clearly separated object in an image needs just a small

Table 6.1: Test Set 1–Comparison of computation time (in seconds) of 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M for image Figure 6.2(a) with different resolutions. Again, the time ratio, t_n/t_{n-1} close to 8.8 indicates $O(N \log N)$ speed while the ratio 2.2 indicates $O(\sqrt[3]{N} \log N)$ speed for that particular test image. All algorithms successfully segmenting the image with additional noise but 3DSC3 is up to 23.3, 29.5 and 1.5 times faster than 3DSC2, 3DSC2M and 3DSC3M respectively.

Algorithm	Size $N = n^3$	JSC	Time, t_n	$\frac{t_n}{t_{n-1}}$
3DSC2	32^3	1.0	17.2	
	64^3	1.0	67.2	3.9
	128^3	1.0	481.5	7.2
	256^3	1.0	3690.6	7.8
3DSC3	32^3	1.0	15.3	
	64^3	1.0	39.1	2.6
	128^3	1.0	70.8	1.8
	256^3	1.0	158.2	2.2
3DSC2M	32^3	1.0	24.3	
	64^3	1.0	84.7	3.5
	128^3	1.0	565.1	6.7
	256^3	1.0	4668.5	8.3
3DSC3M	32^3	1.0	28.3	
	64^3	1.0	61.7	2.2
	128^3	1.0	108.5	1.8
	256^3	1.0	234.9	2.2

θ.

6.8.1 Test Set 1: Segmentation on artificial geometrical objects.

In the first experiment, we compare the performance of 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M in segmenting problem in Figure 6.2(a) with multiple resolutions. The problem is an artificial geometric constraint with some noise added. Here, we take $tol = 10^{-2}$ and $\theta = 30/(n \times 10^{-4})$.

The first column of Figure 6.3 shows successful results from 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M for image size $128 \times 128 \times 128$. The second and third columns show the respective slice representation given by each algorithm. The CPU time needed for our algorithm 3DSC3 is 68.3s (1.1 min) which is about 5, 6.5 and 1.3 times faster than 3DSC2, 3DSC2M and 3DSC3M respectively. The segmentation results for other images size are tabulated in Table 6.1. The ratios of the CPU times in column 5, t_n/t_{n-1} close to 8.8 indicates that 3DSC2 and 3DSC2M are of complexity $O(N \log N)$ while the the ratio 2.2 illustrates the complexity of $O\left(\sqrt[3]{N} \log N\right)$ for both 3DSC3 and 3DSC3M for the particular test image. In addition, 3DSC3 is up to 23.3, 29.5 and 1.5 times faster than 3DSC2, 3DSC2M and 3DSC3M respectively. All algorithms give high segmentation accuracy, indicated by JSC values.

To illustrate the convergence of our multilvel algorithms, we plot the residual of 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M in segmenting image in Figure 6.2(a) for size

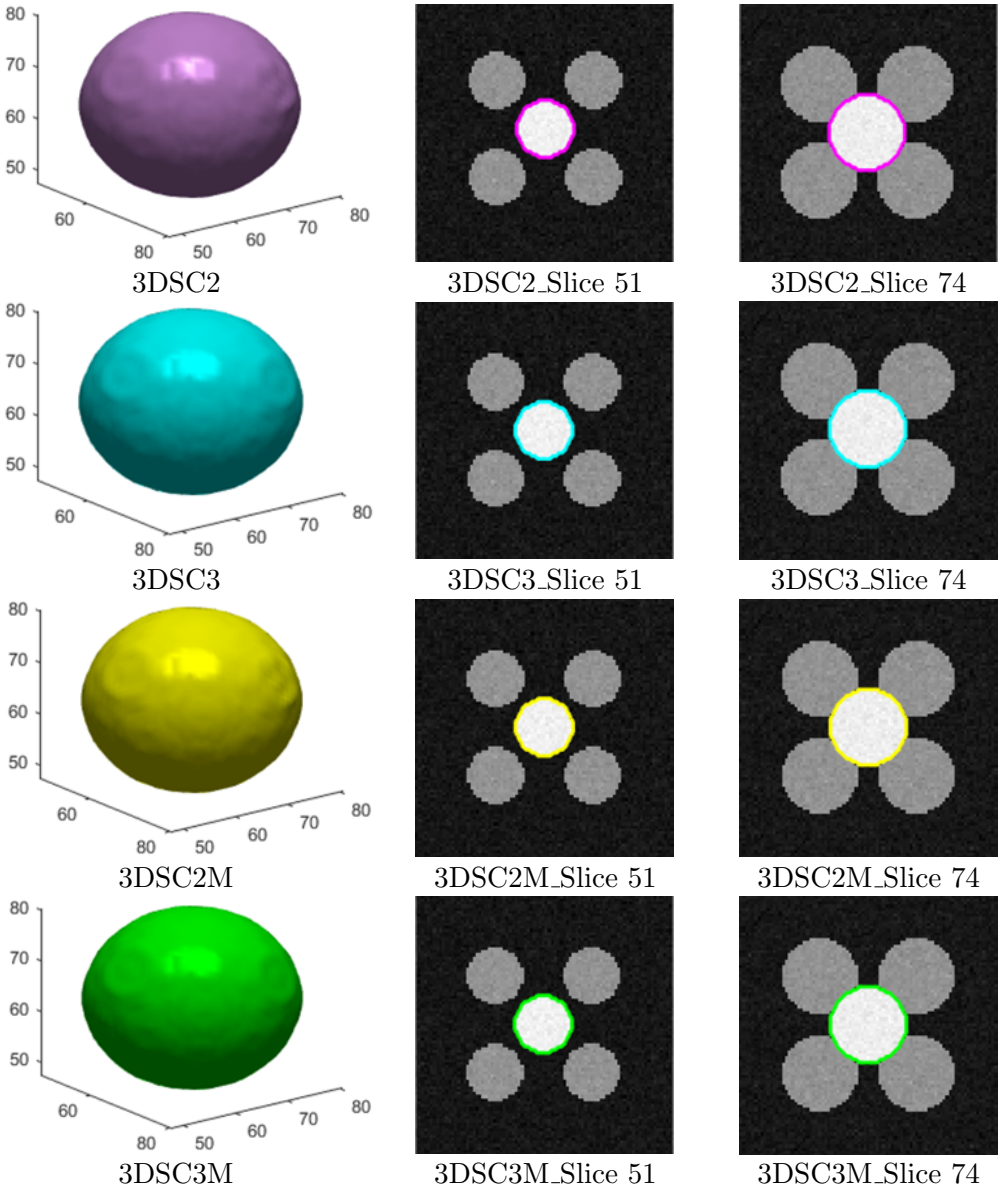


Figure 6.3: Test Set 1 –Successful results from 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M for image size $128 \times 128 \times 128$ in the first column. The second and third columns show the respective slice representation given by each algorithm. The CPU time needed for our algorithm 3DSC3 is 70.8s (1.2 min) which is about 6.8, 8.0 and 1.5 times faster than 3DSC2, 3DSC2M and 3DSC3M respectively .

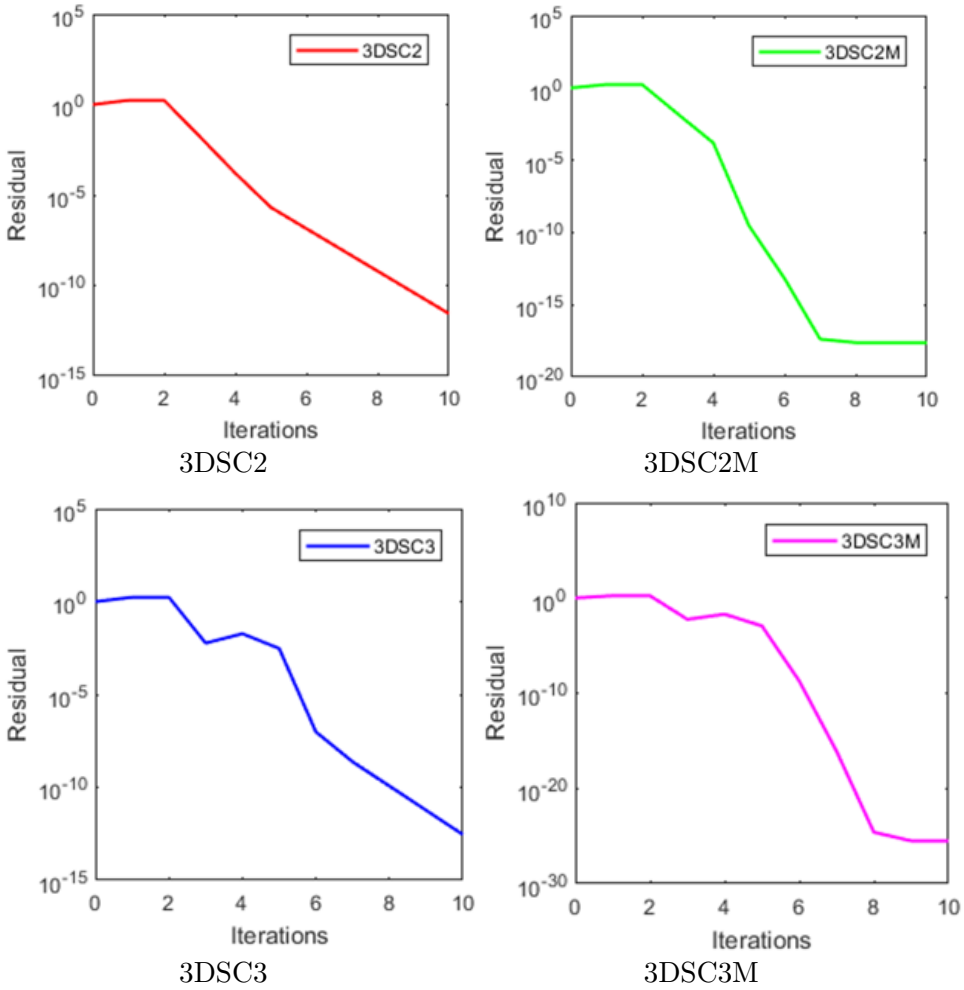


Figure 6.4: Test Set 1–The residual plots for 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M to illustrate the convergence of the algorithms. The extension up to 10 iterations shows that the residual of the algorithms keep reducing.

$64 \times 64 \times 64$ based on Table 6.1. The plots are shown in Figure 6.4. In Figure 6.4, we extend the iteration up to 10 iterations. As we can see, the residual of the algorithms keep reducing.

6.8.2 Test Set 2: Segmentation on real medical data

Next, we test 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M in segmenting medical data in Figure 6.2(b) (blood vessel) and 6.2(c) (kidney). We take $\theta = 11200$ for Figure 6.2(b) and $\theta = 7000$ for Figure 6.2(c). Both images are size of $128 \times 128 \times 128$. The programs stop when $tol = 10^{-2}$. We tabulate the CPU time needed to segment both problems in Table 6.2.

Clearly, 3DSC3 is faster than 3DSC2, 3DSC2M and 3DSC3M. By visual evaluation, Figure 6.5 shows the successful result form 3DSC3 in segmenting both medical images. Figure 6.5(a) and 6.5(b) show the 3-D plots of objects extracted from Figure 6.2(b)

Table 6.2: Test Set 2–The computation time (in seconds) of 3DSC2, 3DSC3, 3DSC2M, and 3DSC3M for segmenting Problem in Figure 6.2(b) (blood vessel) and Figure 6.2(c)(kidney). Notice that 3DSC3 is faster than other algorithms.

Algorithm	Figure 6.2(b)	Figure 6.2(c)
3DSC2	560.1	682.2
3DSC3	265.3	373.6
3DSC2M	702.0	855.3
3DSC3M	396.2	520.7

Table 6.3: Test Set 3–Comparison of computation time (in seconds) of 3DCHB with 3DSC3 for Problem in Figure 6.2(d) for different stopping accuracy. Note 3DSC3 can be up to 56 times faster than 3DCHB for $tol = 10^{-9}$.

Algorithm	Stopping Accuracy	CPU Time	JSC
3DSC3	10^{-1}	8.9	1.0
	10^{-3}	10.3	1.0
	10^{-6}	11.8	1.0
	10^{-9}	13.5	1.0
3DCHB	10^{-1}	12.1	1.0
	10^{-3}	99.2	1.0
	10^{-6}	103.7	1.0
	10^{-9}	757.8	1.0

(blood vessel) and 6.2(c) (kidney) respectively. The slice representation of the extracted object is shown in the second and third columns. Here, we can observed that the targeted object has a small separation gap and has a complex shape especially for kidney case. Notice that all algorithms required more CPU time to segment the larger organ (kidney) compare to the small targeted object (blood vessel).

Based on the experiments in Test Set 1 and Test Set 2, we observe that 3DSC3 performs faster than the 3DSC2, 3DSC2M, and 3DSC3M. Therefore, in practice we recommend 3DSC3 as the better multilevel algorithm for our selective segmentation model.

6.8.3 Test Set 3: Comparison of 3DSC3 with 3DCHB

In this second last experiment, we compare the 3DCHB with 3DSC3 in segmenting an object in Figure 6.2(d) of size $256 \times 256 \times 256$. Here, we used $\theta = 70/(256 \times 10^{-4})$ and vary the stopping accuracy, that is $tol = 10^{-1}, 10^{-3}, 10^{-6}$ and 10^{-9}. The CPU time is recorded for each degree of accuracy. Table 6.3 show the CPU time taken by both algorithms in segmenting the image. One can observe that 3DCHB is almost has similar efficiency with 3DSC3 for $tol = 10^{-1}$. However, as the stopping accuracy is getting smaller, the efficiency of 3DSC3 is better than 3DCHB. This implies that the convergence of 3DCHB is more slower than 3DSC3 when degree of stopping accuracy used is smaller. We visualize the segmentation output from both algorithms using $tol = 10^{-6}$ in Figure 6.6.

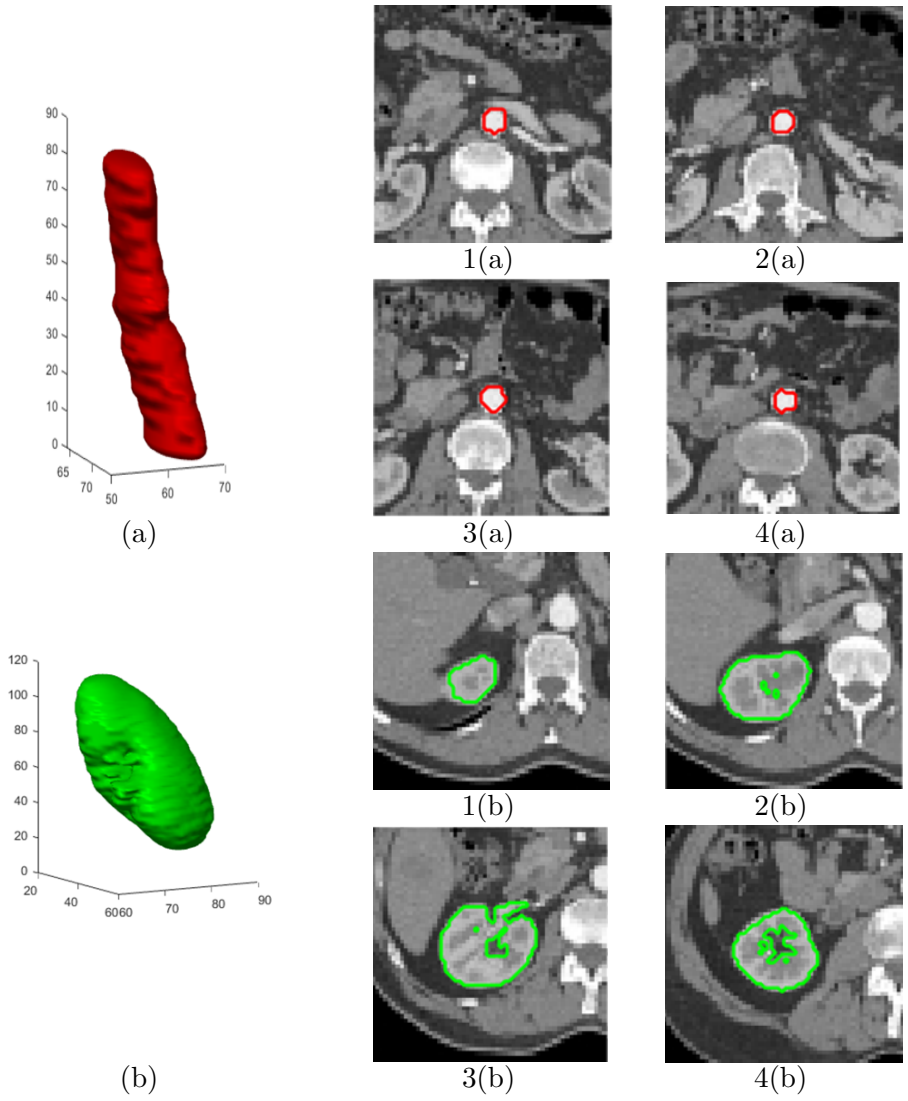


Figure 6.5: Test Set 2–Sucessful segmentation result for 3DSC3 in segmenting medical images. Figure 6.5(a) and 6.5(b) show the 3-D plots of objects extracted from Figure 6.2(b) (blood vessel) and 6.2(c) (kidney) respectively. The figures 1(a)-4(a) and the figures 1(b)-4(b) show the slice representation of the blood vessel and kidney, respectively.

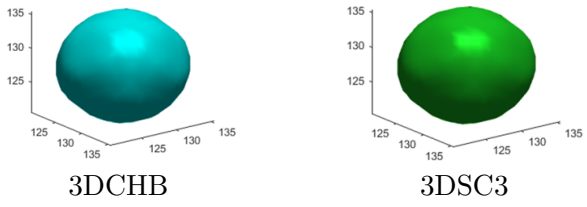


Figure 6.6: Test Set 3–Segmentation of image in Figure 6.2(d) by 3DCHB and 3DSC3 using $tol = 10^{-6}$. 3DSC3 is about 9 times faster than 3DCHB (see Table 6.3).

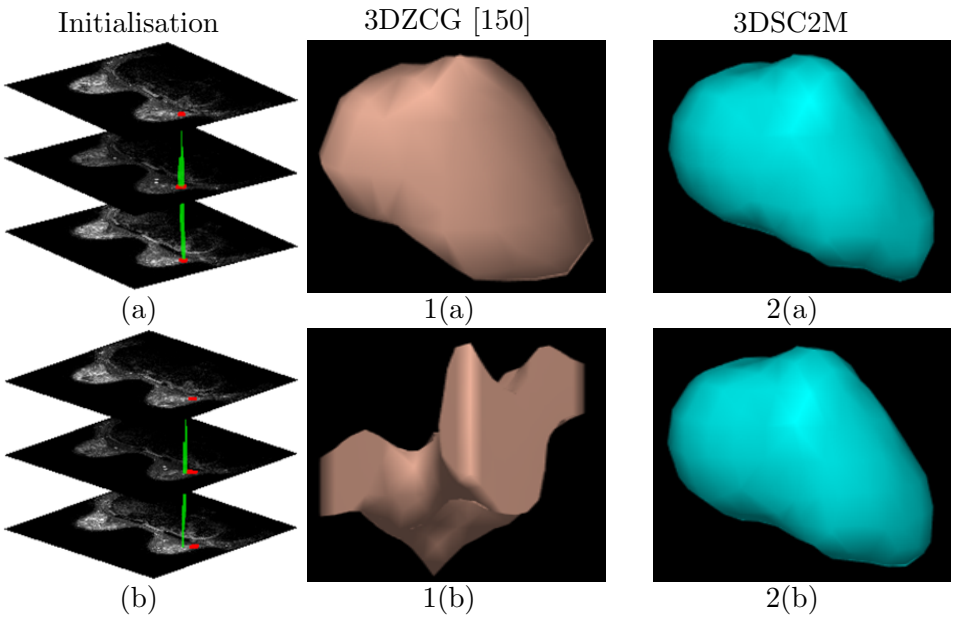


Figure 6.7: Test Set 4–Segmentation performance of 3DZCG and 3DSC3 using 2 different initializations. With initialisation 1 in (a), the segmentation results for 3DZCG and 3DSC3 are illustrated in 1(a) and 2(a) respectively. With initialisation 2 in (b), the segmentation results for 3DZCG and 3DSC3 are illustrated in 1(b) and 2(b) respectively. Clearly, 3DSC3 gives a consistent segmentation result indicating that our 3DSC2M is less dependent on initialisation while 3DZCG is sensitive to initialisation as indicated by different results obtained.

6.8.4 Test Set 4: Comparison of 3DSC3 with 3DZCG [150]

Finally, we compare the performance of our 3DSC3 with 3DZCG [150] in segmenting MRI medical data that contains breast disease using different initialisation. Here, $tol = 10^{-2}$ and $\theta = 10^4$. Figure 6.7(a) shows the data with initial solution is constructed based on the location of marker points while Figure 6.7(b) shows the initial solution is located slightly away from the marker. Figure 6.7 1(a) and 2(a) illustrate the successful result by 3DZCG and 3DSC3 respectively using initial solution in Figure 6.7(a). For the second initialisation in 6.7(b), our 3DSC3 gives a consistent segmentation curve as shown in Figure 6.7 2(b), showing the advantages of our model. However, the segmentation results of 3DZCG are inconsistent which implies that 3DZCG is heavily dependent on the initilization due to highly non-convex terms involve in the minimisation problem.

6.9 Summary

In this chapter, we have successfully developed a 3-D convex and selective segmentation model and solved using a new 3-D optimisation based multilevel algorithm called 3DSC2. Using the local property of a targeted object in an image, we modified the 3DSC2 algorithm and name it as 3DSC3. In order to get a stronger decaying property, new variant for 3DSC2 and 3DSC3 are proposed, called 3DSC2M and 3DSC3M, respectively. Experiments carried out both on artificial geometric images and medical images show

that 3DSC3 performs much faster than 3DSC2, 3DSC2M and 3DSC3M. We also applied Chambolle's projection algorithm to solve our 3-D segmentation model and name it as 3DCHB. Comparison of 3DSC3 with 3DCHB shows that for smaller stopping accuracy value, 3DSC3 performs much faster than 3DCHB. The comparison of 3DSC3 with a recent 3-D selective segmentation model called 3DZCG is also conducted. Results show that 3DSC3 is consistent with different initial solution (less dependent of initialisation) compare to 3DZCG which is sensitive to initial solution. In next work, we will develop a new formulation for higher order selective segmentation model as well as developing multilevel algorithm to solve the model.

Chapter 7

Euler's Elastica based Selective Segmentation Model and Its Fast Algorithm

In this chapter, we propose two new selective segmentation models called ES1 and ES2 that apply the Euler's elastica energy term as boundary regularisation. The energy term provides curvature information of a level curve in an image that may effect the final segmentation boundary. The ES1 model uses relaxed binary curve representation to compute the curvature while ES2 applies level set of continuous signed distance function curve representation to calculate curvature. The minimisation of the objective functions are challenging to solve due to nonsmooth, nonconvex, nonlinear and involves higher order derivatives. We propose a simple and efficient way to solve the Euler's elastica optimisation problems by treating both models as a weighted total variation type selective segmentation problem and we propose multilevel algorithms to solve them. Experimental results show that the ES2 performs better than ES1. Numerical test on ES2 also demonstrated that the model give a satisfactory result especially in segmenting an object with missing or incomplete boundary when compared to other existing models.

7.1 Introduction

One of the most important global image segmentation models is Mumford-Shah model [102], where all features in an image are required to be segmented. The model aims to find a piecewise smooth function which approximates a given image. The methodology has brought forth numerous variational models in many topics of image processing, including segmentation, denoising, and inpainting. An interesting case of the Mumford-Shah's model is the Chan-Vese's model [38]. The Chan-Vese's model [38] approximates a given image using binary piecewise constant representation via level set functions. Chan-Vese model [38] has proved to be an effective global segmentation model in many applications which aim to segment all features in a given image. It also has been

modified in a variety of contexts [37, 82, 138].

The task of segmenting a particular object in an images known as selective segmentation model, is a new subject developed in recent years and not yet been widely studied. As mentioned in the previous chapters, there have been a few variational selective segmentation methods [2, 103, 15, 12, 64, 65, 66, 67, 114, 110, 111, 112, 113, 83, 129, 74].

In summary, Gout-Guyader- Vese [65, 64, 67] firstly proposed an edge-based method for modelling selective segmentation. Later, it was improved by a mixed edge-based and region-based selective model by Badshah-Chen [12]. In literature, there were other approaches such as Random Walks [66], Geodesic [15] and GrabCut [114] that uses the distributions probability, edge-based function or graph-cut theory respectively. Nguyen et al. [103] improved these models by introducing an interactive image segmentation algorithm that is less sensitive to the user inputs and sensitivity to small variations that is able to produce an accurate boundary with a small amount of user interaction. This robust method is considered as the state-of-the-art method for selective segmentation, but with a limitation in segmenting semi-transparent boundaries [103]. The later work by Rada-Chen [110] in formulating dual level set model in 2-D framework as well as in 3-D [113] framework has shown to give satisfactory results for cases where the other methods would produce spurious objects (i.e. fail the selection) in some hard cases where the objects are near to each other or the intensity difference is small. The drawback of the Rada-Chen model [110] is that it is slow to implement, hence they improved the model by introducing one level set formulation in [111]. For the case in dealing with selecting an object with irregular and oscillatory object boundaries, they proposed a variational model for selective image segmentation of features with infinite perimeter [112].

The works of Rada-Chen in [110, 111, 112, 113] are all in non-convex environment, hence sensitive to initial solution. In 2015, Spencer and Chen [129] proposed a convex selective model that more robust in locating initial solution. Recently in 2018, Liu et al. [83] proposed a new convex selective segmentation model based on the work [12] and [102]. In Chapter 5 and [74], we have reformulate the Spencer and Chen [129] model called SC2, so that it is less sensitive to parameters and has cheaper computational cost compared to [129]. The SC2 is also more robust compared to [83] in terms of number of markers used. Moreover, SC2 is able to reduce the limitation of the state-of-the-art model of [103].

The above mentioned selective segmentation models are all grey intensity based models where the segmentation result heavily relies on grey intensity values of the given image. However, due to the complexity of real images, the targeted objects might be occluded by other ones or some parts of them may not be distinguished from the background by the contrast. For example, in medical image analysis, target organs may be blended with other ones, and some parts of them may be occluded by other organs or even missing due to imaging conditions. Therefore, those grey intensity selective based segmentation models might not be well suited to segment those goal objects from given images.

To overcome this issue, one possible way is to incorporate shapes of those targeted objects into the segmentation procedure. Extensive work has also focused on this research topic-segmentation using prior shape knowledge [32, 43, 46, 81, 128]. Another possible way is to overcome this issue is to impose curvature constraints on the segmentation contours to restore those boundaries that are missing or not well defined by the grey intensity gradient of the images. This type of segmentation approach makes use of Euler's elastica energy term in the formulation. It can interpolate the missing boundaries automatically without using prior shape knowledge.

A regular 2-D curve C is said to be Euler's elastica if it is the equilibrium curve of the elasticity energy functional

$$E(C) = \int_C \left(a + b\kappa^2\right) ds \tag{7.1}$$

where ds represents the arc length of the curve C, $\kappa(s)$ the scalar curvature C at point s and a, b are two positive constant parameters. By setting $b = 0$, the energy functional $E(C)$ measures the total length of the curve. If $a = 0$, the $E(C)$ measures the total curvature of the curve.

It was first introduced by Euler in 1744 in studying the physics of thin rods [84]. Euler's elastica energy minimisation model has a long history in image processing. Some of the works using equation (7.1) include segmentation of objects with different depth [53, 104, 155, 134], region-segmentation [156, 51, 119, 120, 132, 14], illusory contour [95, 75, 154, 76] and image denoising and inpainting problems [17, 27, 36, 52, 54, 3, 4, 93, 94, 13, 101, 120, 133, 145, 148, 147, 21].

Let $u(x)$ denotes the gray level of an image u at point x . The level lines are defined as boundaries of upper level sets, define at each gray level q by $X_q = \{x, u(x) \geq q\}$. The Euler's elastica of all level curves of $u(x)$ can be expressed as

$$\int_{q=0}^{Q} \int_{X_q : u=q} \left(a + b\kappa^2\right) ds dq \tag{7.2}$$

where Q is the maximum value of u. Furthermore, in terms of function u , the curvature κ can be expressed as

$$\kappa(u) = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) \tag{7.3}$$

as shown in [13]. Therefore, we can express (7.2) more simply as

$$E(u) = \int_\Omega \left(a + b\left(\nabla \cdot \frac{\nabla u}{|\nabla u|}\right)^2\right) |\nabla u| \, dx \tag{7.4}$$

(see [13, 148] and the reference therein). Fuctional (7.4) represents the Euler's elastica energy of all level curves of u defined on the domain Ω.

For the Chan-Vese type Euler's elastica energy models [156, 132, 14], equation (7.4) is used as the regulariser. The modification allows the models to connect broken or missing part of objects. In addition, it can capture objects of large size while ignoring the objects of small size in a given image. Moreover, the models also less sensitive to

image with intensity inhomogenity compare to the original Chan-Vese model [38]. Note that, if $b = 0$, (7.4) is reduced to TV regulariser.

The segmentation models above ([156, 132, 14]) are global segmentation. Although they have ability to capture object of large size and ignoring the ones with smaller size, the selective segmentation model is still need to be developed in case if the targeted object (foreground) in the given image is smaller than the others (background). To this end, in this chapter, we will develop two new selective segmentation model called ES1 and ES2 that employ Euler's elastica in (7.4) as a new regularisation.

The minimisation of the resulting objective functions for both ES1 and ES2 are challenging to solve due to nonsmooth, nonconvex, nonlinear and involves higher-order derivatives. We will show a simple and efficient way to solve the Euler's elastica selective segmentation problems by treating the problems as a weighted total variation type selective segmentation problems, inspired by the related work in [132, 148, 14]. We solved the models using optimisation-based multilevel algorithm developed in Chapter 4 and 5.

We conclude the introduction part with the organisation of this chapter. In Section 7.2 we first review the global Chan-Vese Euler's elastica-based segmentation models. We propose two new formulations of selective segmentation models based on Euler's elastica energy called ES1 and ES2 in Section 7.3. Then, we develop the multilevel algorithms to solve both models in Section 7.4. Section 7.5 presents numerical experiments conducted to determine the better selective segmentation model. Then, the chosen model is compared with other models. We conclude this chapter in Section 7.6.

7.2 Review on Euler's elastica based Chan-Vese's global segmentation model

Recall from Section 3.34 that the Chan-Vese model [38] is written as

$$\begin{aligned} &\min_{\phi,c_1,c_2} F_{CV}\left(\phi, c_1, c_2\right), \\ &F_{CV}\left(\phi, c_1, c_2\right) = \mu \int_{\Omega} |\nabla H(\phi)| \, d\Omega + \lambda_1 \int_{\Omega} (z - c_1)^2 H(\phi) \, d\Omega \\ &\qquad\qquad + \lambda_2 \int_{\Omega} (z - c_2)^2 (1 - H(\phi)) \, d\Omega \end{aligned} \tag{7.5}$$

Here, the level set function ϕ is defined as a signed distance function from the unknown segmentation boundary Γ where outside Γ the sign of ϕ is negative while inside Γ the sign of ϕ is positive. The zero level set function ϕ is used to represent Γ that is $\Gamma = \{(x, y) \in \Omega : \phi(x, y) = 0\}$. The function $H(\phi)$ is the Heaviside function of ϕ , the constants c_1 and c_2 are viewed as the average values of a given image z inside and outside the unknown curve Γ. The fixed parameters μ , λ_1 , and λ_2 are positive parameters. The Chan-Vese model [38] has proved to be an effective global segmentation model. As discussed in Chapter 3, the model also has been modified in variety of context such as multiphase segmentation [138], vector-valued image segmentation [37], and selective image segmentation [12, 111].

However, the length regularisation term is insufficient to segment an image with missing or broken boundary. In 2003, Zhu et al. [156] proposed to modify the Chan-Vese model [38] by employing Euler's elastica energy term as

$$\begin{aligned}
&\min_{\phi,c_1,c_2} E_{CVE}\left(\phi,c_1,c_2\right),\\
&E_{CVE}\left(\phi,c_1,c_2\right)=\int_\Omega\left(a+b\left|\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\right|^2\right)|\nabla H(\phi)|\,d\Omega+\lambda_1\int_\Omega(z-c_1)^2H(\phi)\,d\Omega\\
&\qquad\qquad+\lambda_2\int_\Omega(z-c_2)^2(1-H(\phi))\,d\Omega
\end{aligned}\tag{7.6}$$

where a and b are positive parameters. The parameter a controls the length term and b controls the curvature of the segmentation boundary. Here, the length term of the Chan-Vese model [38] is still needed to avoid excessive segmentation contours.

To reduce the complexity of the model (7.6), Zhu et al [156] relaxed the model as done in [34] and proposed to solve

$$\begin{aligned}
&\min_{u\in[0,1]} E_{CVE}\left(u,c_1,c_2\right),\\
&E_{CVE}\left(u,c_1,c_2\right)=\int_\Omega\left(a+b\left|\nabla\cdot\frac{\nabla u}{|\nabla u|}\right|^2\right)|\nabla u|\,d\Omega+\lambda_1\int_\Omega(z-c_1)^2u\,d\Omega\\
&\qquad\qquad+\lambda_2\int_\Omega(z-c_2)^2(1-u)\,d\Omega
\end{aligned}\tag{7.7}$$

Experiments showed that the model [156], solved using Augmented Lagrangian Method (ALM) is able to connect broken parts of objects to form meaningful objects. In addition, the model can capture objects of large size while ignoring the ones of small size [156]. Note that the work from [156] employ the L^2-Euler's elastica energy. There is also work from [14] that employ the L^1-Euler's elastica energy for image segmentation and the comparisons are made in [70].

Due to difficulty in solving Euler's elastica energy related models that involve highly nonlinear and higher order terms, Tai and Duan [132] proposed to combine both functionals (7.6) and (7.7) and produced the following model

$$\begin{aligned}
&\min_{u,\phi} E_M\left(u,\phi,c_1,c_2\right),\\
&E_M\left(u,\phi,c_1,c_2\right)=\int_\Omega g(\phi)\,|\nabla u|\,d\Omega+\lambda_1\int_\Omega(z-c_1)^2u\,d\Omega\\
&\qquad\qquad+\lambda_2\int_\Omega(z-c_2)^2(1-u)\,d\Omega
\end{aligned}\tag{7.8}$$

under constraint:

$$u=H(\phi),$$

with

$$g(\phi) = a + b\left|\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}\right|^2. \tag{7.9}$$

The functional (7.8) is viewed as a weighted total variation type which is more easy and efficient to solve compared to functionals (7.6) and (7.7) [132]. Here, the binary function u is used to express the length term and the signed distance function (SDF) ϕ to compute the curvature values. By representing a curve using the SDF function ϕ, the geometric features of the curve such as normal and curvature can be computed more naturally than the relaxed binary representation used in (7.6) [132].

The minimisation of (7.8) involves two unknowns u and ϕ. Hence, the alternating minimisation procedure is done to solve it. First, ϕ is fixed and solution for u is obtained by minimising the following functional

$$\begin{aligned} u = \underset{u\in\{0,1\}}{\arg\min} \int_\Omega g(\phi)\,|\nabla u|\; d\Omega + \lambda_1 \int_\Omega (z-c_1)^2 u\, d\Omega \\ +\lambda_2 \int_\Omega (z-c_2)^2 (1-u)\, d\Omega. \end{aligned} \tag{7.10}$$

The functional (7.10) is relaxed following the idea by Chan et al in [34] and it is equivalent to the following problem:

$$\begin{aligned} u = \underset{u\in[0,1]}{\arg\min} \int_\Omega g(\phi)\,|\nabla u|\; d\Omega + \lambda_1 \int_\Omega (z-c_1)^2 u\, d\Omega \\ +\lambda_2 \int_\Omega (z-c_2)^2 (1-u)\, d\Omega \end{aligned} \tag{7.11}$$

in the sense that a threshold of a solution u in (7.11) by a value $\eta \in (0,1)$ (usually $\eta = 0.5$) gives a minimiser for (7.10). They adopted the Split Bregman method [63, 48, 49, 50] to solve (7.11).

After u is obtained, then u is fixed to solve for ϕ from

$$u = H(\phi).$$

Given a binary function u, there is a unique SDF of ϕ satisfying the above relation that is denoted by

$$\phi = SDF(u) \tag{7.12}$$

as shown by [132]. The function ϕ in (7.12) is iteratively solved by re-distance process using the fast sweep method [136, 153] to the η-level set of u. Fast marching methods [122, 123] are also suggested to solve (7.12). The alternating minimisation procedure above is done until convergence of the system achieved. The efficiency and effectiveness of the proposed approach are validated through extensive experiments done.

In the same paper, Tai and Duan [132] also minimised the Euler's elastica based image inpainting model, and illusory shape reconstruction model by simplifying the problem as a weighted TV based model. Papers in other image processing applications

from [148] and [13] also used the similar idea which simplifies the Euler'elastica based image registration model and Euler'elastica based image denoising model into a weighted TV-type registration model and denoising model, respectively.

7.3 Euler's elastica based selective segmentation model

Selective image segmentation can be defined as the task of extracting one object of interest in an image based on some additional information of geometric constraints. The segmentation models that we have reviewed in the previous section [156, 132, 14] are for global segmentation due to the fact that all features or objects in an image are to be segmented. Although they have the ability to segment an object of large size and ignore the other objects of smaller size in a given image, the selective segmentation model is still need to be developed in case the targeted object in a given image is smaller than the other objects.

The selective segmentation models of [12, 111, 129] based Chan-Vese have proven to be effective. As these models are based on Chan-Vese formulation [38] that heavily relies on grey intensity values of a given image and length regularisation term, they are not well suited to segment images with incomplete boundary or object that might be occluded by other objects. To perform the job, we will proposed two new formulations of selective segmentation models that employ Euler's elastica energy and we will call them ES1 and ES2. This energy gives information about the image curvature which is hope that both ES1 and ES2 can accomplish the task.

As in Chapters 4 and 5, let n_1 geometric constraints be given by a marker set

$$A = \{w_i = (x_i^*, y_i^*) \in \Omega, 1 \leq i \leq n_1\} \subset \Omega$$

where each point is near the object boundary Γ, not necessarily on it [111, 152]. The selective segmentation idea tries to detect the boundary of a single object among all homogeneity intensity objects in image domain Ω close to A; here $n_1\,(\geq 3)$. The geometrical points in A define an initial polygonal contour and guide its evolution towards Γ [152].

Using the set A, construct a polygon Q that connects up the markers. Denote the function $P_d\,(x, y)$ (as used in [129]) the normalized Euclidean distance of each point $(x, y) \in \Omega$ from its nearest point $(x_p, y_p) \in Q$:

$$P_d\,(x, y) = \frac{P_0\,(x, y)}{\|P_0\|_{L^\infty}}, \qquad P_0\,(x, y) = \sqrt{(x - x_p)^2 + (y - y_p)^2} = \min_{q \in Q} \|(x, y) - (x_q, y_q)\|.$$

Then the ES1 formulation minimising a cost function defined as follows

$$\begin{aligned}
&\min_{u\in[0,1]} ES1\,(u)\,,\\
&ES1\,(u) = \int_\Omega \left(a + b\left|\nabla\cdot\frac{\nabla u}{|\nabla u|}\right|^2\right)|\nabla u|\;d\Omega + \lambda_1\int_\Omega (z-c_1)^2 u\,d\Omega \\
&\qquad +\lambda_2\int_\Omega (z-c_2)^2(1-u)\,d\Omega + \theta\int_\Omega P_d u\,d\Omega
\end{aligned} \tag{7.13}$$

Note that if $\theta = 0$, (7.13) reduces to (7.7) of [156]. The functional (7.13) which involves Euler's elastica energy term is challenging to solve due to nonconvexity, nonlinearity and involves higher order derivatives. As explored in some applications previously mentioned such as [132, 13, 148], it is often more efficient to consider computing the curvature term separately from an optimisation functional. Motivating from the work in [132, 13, 148], the curvature term in (7.13) is computed separately and we solve the Euler's elastica selective segmentation model (7.13) as a weighted TV based optimisation model and proposed to solve the unconstraint minimisation as follows

$$\begin{aligned}
&\min_{u} ES1\,(u)\,,\\
&ES1\,(u) = \int_\Omega V(\kappa)\;|\nabla u|\;d\Omega + \lambda_1\int_\Omega (z-c_1)^2 u\,d\Omega + \lambda_2\int_\Omega (z-c_2)^2(1-u)\,d\Omega \\
&\qquad +\theta\int_\Omega P_d u\,d\Omega + \alpha\int_\Omega \nu(u)\,d\Omega
\end{aligned} \tag{7.14}$$

with

$$\begin{aligned}
V(\kappa) &= a + b|\kappa|^2,\\
\kappa(u) &= \nabla\cdot\frac{\nabla(u)}{|\nabla(u)|}
\end{aligned} \tag{7.15}$$

The term $\nu(u)$ in (7.14) enforces the solution u to be constrained in $[0,1]$. One may consider using the formula for $\nu(u)$ as in [129]. However, it increases the computational complexity of the model. We propose to use splitting or alternating minimisation approach idea as used in Chapter 5. Now, by introducing an artificial variable w, (7.14) takes the following form

$$\begin{aligned}
&\min_{u,w} ES1\,(u,w)\,,\\
&ES1\,(u,w) = \int_\Omega V(\kappa)\;|\nabla u|\;d\Omega + \lambda_1\int_\Omega (z-c_1)^2 w\,d\Omega + \lambda_2\int_\Omega (z-c_2)^2(1-w)\,d\Omega \\
&\qquad +\theta\int_\Omega P_d w\,d\Omega + \alpha\int_\Omega \nu(w)\,d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2 d\Omega
\end{aligned} \tag{7.16}$$

The right-most term in (7.16) enforces $w \approx u$ for sufficiently small $\rho > 0$ as used in [20, 40, 99]. The minimisation problem (7.16) now has two unknowns and we shall use

an alternating minimisation-like procedure to solve it. First, we fix w and solve the following functional

$$\begin{aligned} &\min_{u} ES1^{1a}(u,w), \\ &ES1^{1a}(u,w) = \int_{\Omega} V(\kappa)\ |\nabla u|\ d\Omega + \frac{1}{2\rho}\int_{\Omega}(u-w)^2 d\Omega. \end{aligned} \tag{7.17}$$

We propose to solve (7.17) using optimisation based multilevel algorithm that we developed in Chapters 4 and 5. The implementation details are discussed in the next section.

Then u is fixed to solve

$$\begin{aligned} &\min_{w\in[0,1]} ES1^{1b}(u,w), \\ &ES1^{1b}(u,w) = \lambda_1\int_{\Omega}(z-c_1)^2 w\, d\Omega + \lambda_2\int_{\Omega}(z-c_2)^2(1-w)\, d\Omega \\ &\qquad +\theta\int_{\Omega} P_d w\, d\Omega + \frac{1}{2\rho}\int_{\Omega}(u-w)^2 d\Omega. \end{aligned} \tag{7.18}$$

Functional (7.18) can be solved directly and the analytical solution is defined as

$$w = \min\{\max\{u - \rho(\lambda_1(z-c_1)^2 - \lambda_2(z-c_2)^2) - \rho\theta P_d, 0\}, 1\}. \tag{7.19}$$

After u and w are updated from (7.17) and (7.19) respectively, we go back to update $V(\kappa)$ from (7.15) again. This procedure is repeated until convergence is achieved. As a common practice, the final segmented image is defined by $\Sigma = \{(x,y) : u(x,y) \geq \eta\}$ where the threshold value $\eta \in (0,1)$, usually $\eta = 0.5$.

The ES1 model above uses the relaxed binary curve representation to compute curvature. Another approach to evaluate curvature is to use the level set of continuous SDF. Let ϕ to be the SDF, the formulation of our second selective segmentation based Euler's elastica called ES2 is given as

$$\begin{aligned} &\min_{u\in\{0,1\},\phi} ES2(u,\phi), \\ &ES2(u) = \int_{\Omega}\left(a + b\left|\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\right|^2\right)|\nabla u|\ d\Omega + \lambda_1\int_{\Omega}(z-c_1)^2 u\, d\Omega \\ &\qquad +\lambda_2\int_{\Omega}(z-c_2)^2(1-u)\, d\Omega + \theta\int_{\Omega} P_d u\, d\Omega \end{aligned} \tag{7.20}$$

under constraint:

$$u = H(\phi).$$

We remark that if $\theta = 0$, the ES2 reduces to the model of [132]. To solve ES2, we shall try to use similar ideas as were used for ES1 where the curvature term in (7.20) is computed separately and we solve the Euler's elastica selective segmentation model

(7.20) as a weighted TV based optimisation model and proposed to solve

$$\begin{aligned}
&\min_{u\in\{0,1\},\phi} ES2\,(u,\phi)\,,\\
&ES2\,(u) = \int_\Omega S(\kappa)\ |\nabla u|\ d\Omega + \lambda_1 \int_\Omega (z-c_1)^2 u\, d\Omega + \theta \int_\Omega P_d u\, d\Omega\\
&\qquad +\lambda_2 \int_\Omega (z-c_2)^2 (1-u)\, d\Omega
\end{aligned} \tag{7.21}$$

such that $u = H(\phi)$ with

$$\begin{aligned}
S(\kappa) &= a + b|\kappa|^2,\\
\kappa(\phi) &= \nabla \cdot \frac{\nabla(\phi)}{|\nabla(\phi)|}.
\end{aligned} \tag{7.22}$$

Following further the work of [132], alternating minimisation procedure to solve (7.21) is used. Let us denote $r = (z-c_1)^2 - (z-c_2)^2$.

Firstly, we fix ϕ in (7.21) and solve

$$\min_{u\in\{0,1\}} \left\{ ES2\,(u,\phi) = \int_\Omega S\,(\kappa)\ |\nabla u|\ d\Omega + \int_\Omega ru\, d\Omega + \theta \int_\Omega P_d u\, d\Omega. \right\} \tag{7.23}$$

It is known from [34] that functional (7.23) is equivalent to the following problem

$$\min_{u\in[0,1]} \left\{ ES2\,(u,\phi) = \int_\Omega S\,(\kappa)\ |\nabla u|\ d\Omega + \int_\Omega ru\, d\Omega + \theta \int_\Omega P_d u\, d\Omega \right\} \tag{7.24}$$

where a threshold of a solution (7.24) by a value $\eta \in (0,1)$ gives a minimiser for (7.23). That means the segmented image is defined by $\Sigma = \{(x,y) : u(x,y) \geq \eta\}$. Like ES1, we change (7.24) in unconstrained form and introduce an auxiliary variable w as follows

$$\begin{aligned}
&\min_{u,w} ES2\,(u,w,\phi)\\
&ES2\,(u,w,\phi) = \int_\Omega S\,(\kappa)\ |\nabla u|\ d\Omega + \int_\Omega rw\, d\Omega + \theta \int_\Omega P_d w\, d\Omega\\
&+\alpha \int_\Omega \nu(w)\, d\Omega + \frac{1}{2\rho} \int_\Omega (u-w)^2\, d\Omega
\end{aligned} \tag{7.25}$$

To solve (7.25), we fix w and solve

$$\min_u \left\{ ES2^{2a}\,(u,w,\phi) = \int_\Omega S\,(\kappa)\ |\nabla u|\ d\Omega + \frac{1}{2\rho} \int_\Omega (u-w)^2\, d\Omega \right\} \tag{7.26}$$

using multilevel algorithm that will be discussed in next section. Then, we fix u and solve

$$\min_{w\in[0,1]} \left\{ ES2^{2b}\,(u,w,\phi) = \int_\Omega rw\, d\Omega + \theta \int_\Omega P_d w\, d\Omega + \frac{1}{2\rho} \int_\Omega (u-w)^2\, d\Omega \right\} \tag{7.27}$$

which has an analytical solution as follows

$$w = \min\{\max\{u - \rho r - \rho\theta P_d, 0\}, 1\}. \tag{7.28}$$

Secondly, we fix u in (7.21) to solve for ϕ. Following the work from [132], ϕ is updated using

$$\phi = SDF\,(u)\,. \tag{7.29}$$

The solution for (7.29) is computed using any reinitialisation procedure. In this work we employ Sussman method [131] as discussed in Section 3.34 of Chapter 3. The reinitialisation of ϕ done in (7.29) to be the SDF of η-level set u obtained from (7.26) prevents the level set function of ϕ from becoming too "flat" which causes inaccuracy in computing the curvature term.

After the solutions u, w, and ϕ are updated from (7.26), (7.28), and (7.29) respectively, we go back to update $S(\kappa)$ in (7.22). This iteration is repeated until the system converges.

7.4 Minimisation using optimisation based multilevel algorithm

An optimisation based multilevel method is based on a discretise-optimise scheme where minimisation is solved directly (without using PDEs). This method is an alternative to multigrid method where the multigrid method is based on an optimise-discretise scheme which solve the resulting Euler Lagrange PDE derived from the variational problem. The original multilevel method applied to image denoising [33] used the "patch detection" idea. However, as image size increases, the method can be slow because of the patch detection idea searches the entire image for the possible patch size on the finest level after each multilevel cycle [73].

This section proposes to develop a multilevel algorithm without the "patch detection" idea for our new Euler's elastica based selective segmentation models: first the ES1 model described in (7.14) and then the ES2 model in (7.21).

As detailed in Chapter 4 and 5, we shall assume $n = 2^L$ for a given image z of size $n \times n$. The standard coarsening defines $L + 1$ levels: $k = 1$ (finest) $, 2, ..., L,\ L + 1$ (coarsest) such that level k has $\tau_k \times \tau_k$ "superpixels" with each "superpixels" having pixels $b_k \times b_k$ where $\tau_k = n/2^{k-1}$ and $b_k = 2^{k-1}$, see Figure 4.2 in Chapter 4.

7.4.1 Computation of $V(\kappa(u))$ and $S(\kappa(\phi))$ terms in discrete form

The terms $V(\kappa(u))$ in (7.15) and $S(\kappa(\phi))$ in (7.22) contain a curvature, κ. We first show the computation of $\kappa(u)$ in (7.15) of ES1 in discrete form. Let $\Omega = \{\,(i,j)|\,1 \leq i \leq n, 1 \leq j \leq n\}$ be the discretised image domain and each point (i,j) correspond to a pixel point of the image. The discrete gradient $\nabla u = \left(u^x_{i,j}, u^y_{i,j}\right)$ is given by the central difference scheme

$$u^x_{i,j} = \begin{cases} (u_{i+1,j} - u_{i-1,j})/2 & \text{if} \quad 1 < i < n \\ u_{2,j} - u_{1,j} & \text{if} \quad i = 1 \\ u_{n,j} - u_{n-1,j} & \text{if} \quad i = n \end{cases}$$
$$u^y_{i,j} = \begin{cases} (u_{i,j+1} - u_{i,j-1})/2 & \text{if} \quad 1 < j < n \\ u_{i,2} - u_{i,1} & \text{if} \quad j = 1 \\ u_{i,n} - u_{i,n-1} & \text{if} \quad j = n. \end{cases}$$

We thus give the discretisation of $\kappa(u)$ in (7.15) of ES1 by

$$\kappa_{i,j}(u_{i,j}) = \left(\frac{u^x_{i,j}}{\sqrt{\left(u^x_{i,j}\right)^2 + \left(u^y_{i,j}\right)^2 + \varepsilon}} \right)^x + \left(\frac{u^y_{i,j}}{\sqrt{\left(u^x_{i,j}\right)^2 + \left(u^y_{i,j}\right)^2 + \varepsilon}} \right)^y.$$

Consequently, the term $V(\kappa(u))$ in (7.15) is compute in discrete form as follow

$$V_{i,j}(\kappa_{i,j}(u_{i,j})) = a + b|\kappa_{i,j}(u_{i,j})|^2 \tag{7.30}$$

Similarly, the discretisation of $\kappa(\phi)$ in (7.22) of ES2 is defined by

$$\kappa_{i,j}(\phi_{i,j}) = \left(\frac{\phi^x_{i,j}}{\sqrt{\left(\phi^x_{i,j}\right)^2 + \left(\phi^y_{i,j}\right)^2 + \varepsilon}} \right)^x + \left(\frac{\phi^y_{i,j}}{\sqrt{\left(\phi^x_{i,j}\right)^2 + \left(\phi^y_{i,j}\right)^2 + \varepsilon}} \right)^y.$$

Hence, the computation of $S(\kappa(\phi))$ in (7.22) in discrete form is given as

$$S_{i,j}(\kappa_{i,j}(\phi_{i,j})) = a + b|\kappa_{i,j}(\phi_{i,j})|^2 \tag{7.31}$$

where $\varepsilon > 0$ is a relatively small value based on machine epsilon is added to make the problem well-posed, i.e., to avoid division by zero in the evaluation of curvature.

7.4.2 A multilevel algorithm for ES1

The minimiser for the proposed ES1 model in (7.14) shall be found by iterating (7.17), (7.18), and (7.15) until the system converges. Note that, the solution of (7.18) is obtained analytically following (7.19) while (7.15) (equivalent to (7.30)) is updated by the most recent update of u from (7.17). Now, it remains for us to develop a multilevel algorithm to solve (7.17).

Firstly, we discretise the functional $ES1^{1a}(u, w)$ of problem (7.17) as follows:

$$\begin{aligned} &\min_u ES1^{1a}(u, V, w) \equiv \min_u ES1^{1a}_\alpha(u_{1,1}, u_{2,1}, ..., u_{i-1,j}, u_{i,j}, u_{i+1,j}, ..., u_{n,n}, V, w) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} V_{i,j}\sqrt{(u_{i,j} - u_{i,j+1})^2 + (u_{i,j} - u_{i+1,j})^2 + \beta} + \frac{1}{2\rho}\sum_{i=1}^{n}\sum_{j=1}^{n}(u_{i,j} - w_{i,j})^2 \end{aligned} \tag{7.32}$$

Consider the local minimisation of (7.32) by the coordinate descent method on the finest level 1:

$$\begin{cases} \text{Given} & u^{(0)} = \left(u_{i,j}^{(0)}\right) \text{ with } m = 0, \\ \text{Solve} & u_{i,j}^{(m+1)} = \underset{u_{i,j}\in\mathbb{R}}{\arg\min}\, ES1_{loc}^{1a}\left(u, V, w\right) \text{ for } i,j = 1,2,...n, \\ & \text{Repeat the above steps with } m = m+1 \text{ until stopped.} \end{cases} \tag{7.33}$$

where

$$\begin{aligned} ES_{loc}^{1a} &\equiv ES1_{\alpha}^{1a} - ES1_{\bar{\alpha}}^{1a} \\ &= V_{i,j}\sqrt{\left(u_{i,j} - u_{i+1,j}^{(m)}\right)^2 + \left(u_{i,j} - u_{i,j+1}^{(m)}\right)^2 + \beta} \\ &+ V_{i-1,j}\sqrt{\left(u_{i,j} - u_{i-1,j}^{(m)}\right)^2 + \left(u_{i-1,j}^{(m)} - u_{i-1,j+1}^{(m)}\right)^2 + \beta} \\ &+ V_{i,j-1}\sqrt{\left(u_{i,j} - u_{i,j-1}^{(m)}\right)^2 + \left(u_{i,j-1}^{(m)} - u_{i+1,j-1}^{(m)}\right)^2 + \beta} \\ &+ \tfrac{1}{2\rho}(u_{i,j} - w_{i,j})^2. \end{aligned}$$

The terms $ES1_{\bar{\alpha}}^{1a}$ refers to a collection of all terms that are not dependent on $u_{i,j}$. For $u_{i,j}$ at the boundary (the total variation term), Neumann's condition is used. Minimisation of $ES1_{loc}^{1a}$ in (7.33) is achieved using Newton method by iterating $u^{(m)} \to u \to u^{(m+1)}$:

$$\begin{aligned} &\frac{V_{i,j}\left(2u_{i,j} - u_{i+1,j}^{(m)} - u_{i,j+1}^{(m)}\right)}{\sqrt{\left(u_{i,j} - u_{i+1,j}^{(m)}\right)^2 + \left(u_{i,j} - u_{i,j+1}^{(m)}\right)^2 + \beta}} + \frac{V_{i-1,j}\left(u_{i,j} - u_{i-1,j}^{(m)}\right)}{\sqrt{\left(u_{i,j} - u_{i-1,j}^{(m)}\right)^2 + \left(u_{i-1,j}^{(m)} - u_{i-1,j+1}^{(m)}\right)^2 + \beta}} \\ &+ \frac{V_{i,j-1}\left(u_{i,j} - u_{i,j-1}^{(m)}\right)}{\sqrt{\left(u_{i,j} - u_{i,j-1}^{(m)}\right)^2 + \left(u_{i,j-1}^{(m)} - u_{i+1,j-1}^{(m)}\right)^2 + \beta}} + \tfrac{1}{\rho}\left(u_{i,j} - w_{i,j}\right) = 0 \end{aligned}$$

giving rise to the form

$$u_{i,j}^{new} = u_{i,j}^{old} - T^{old}/B^{old} \tag{7.34}$$

where

$$\begin{aligned} T^{old} &= \frac{V_{i,j}\left(2u_{i,j}^{old} - u_{i+1,j}^{(m)} - u_{i,j+1}^{(m)}\right)}{\sqrt{\left(u_{i,j}^{old} - u_{i+1,j}^{(m)}\right)^2 + \left(u_{i,j}^{old} - u_{i,j+1}^{(m)}\right)^2 + \beta}} + \frac{V_{i-1,j}\left(u_{i,j}^{old} - u_{i-1,j}^{(m)}\right)}{\sqrt{\left(u_{i,j}^{old} - u_{i-1,j}^{(m)}\right)^2 + \left(u_{i-1,j}^{(m)} - u_{i-1,j+1}^{(m)}\right)^2 + \beta}} \\ &+ \frac{V_{i,j-1}\left(u_{i,j}^{old} - u_{i,j-1}^{(m)}\right)}{\sqrt{\left(u_{i,j}^{old} - u_{i,j-1}^{(m)}\right)^2 + \left(u_{i,j-1}^{(m)} - u_{i+1,j-1}^{(m)}\right)^2 + \beta}} + \tfrac{1}{\rho}\left(u_{i,j}^{old} - w_{i,j}\right) \end{aligned}$$

$$
\begin{aligned}
B^{old} = & \frac{2V_{i,j}}{\sqrt{\left(u_{i,j}^{old}-u_{i+1,j}^{(m)}\right)^2+\left(u_{i,j}^{old}-u_{i,j+1}^{(m)}\right)^2+\beta}} - \frac{V_{i,j}\left(2u_{i,j}^{old}-u_{i+1,j}^{(m)}-u_{i,j+1}^{(m)}\right)^2}{\sqrt{\left(\left(u_{i,j}^{old}-u_{i+1,j}^{(m)}\right)^2+\left(u_{i,j}^{old}-u_{i,j+1}^{(m)}\right)^2+\beta\right)^{\frac{3}{2}}}} \\
& + \frac{V_{i-1,j}}{\sqrt{\left(u_{i,j}^{old}-u_{i-1,j}^{(m)}\right)^2+\left(u_{i-1,j}^{(m)}-u_{i-1,j+1}^{(m)}\right)^2+\beta}} - \frac{V_{i-1,j}\left(u_{i,j}^{old}-u_{i-1,j}^{(m)}\right)^2}{\sqrt{\left(\left(u_{i,j}^{old}-u_{i-1,j}^{(m)}\right)^2+\left(u_{i-1,j}^{(m)}-u_{i-1,j+1}^{(m)}\right)^2+\beta\right)^{\frac{3}{2}}}} \\
& + \frac{V_{i,j-1}}{\sqrt{\left(u_{i,j}^{old}-u_{i,j-1}^{(m)}\right)^2+\left(u_{i,j-1}^{(m)}-u_{i+1,j-1}^{(m)}\right)^2+\beta}} - \frac{V_{i,j-1}\left(u_{i,j}^{old}-u_{i,j-1}^{(m)}\right)^2}{\sqrt{\left(\left(u_{i,j}^{old}-u_{i,j-1}^{(m)}\right)^2+\left(u_{i,j-1}^{(m)}-u_{i+1,j-1}^{(m)}\right)^2+\beta\right)^{\frac{3}{2}}}} \\
& + \frac{1}{\rho}.
\end{aligned}
$$

To introduce the multilevel algorithm, it is of interest to rewrite (7.33) in an equivalent form:

$$
\begin{cases}
\text{Given} & u^{(0)} = \left(u_{i,j}^{(0)}\right) \quad \text{with } m = 0, \\
\text{Solve} & \hat{c} = \underset{u_{i,j}\in\mathbb{R}}{\arg\min}\, ES1_{loc}^{1a}\left(u_{i,j}^{(m)} + c, V, w\right), \\
\text{Update} & u_{i,j}^{(m+1)} = u_{i,j}^{(m)} + \hat{c}, \\
& \text{Repeat the above steps with } m = m+1 \text{ until stopped.}
\end{cases}
\tag{7.35}
$$

The problem in (7.35) can be interpreted as finding the best correction constant $\hat{c}$ at the current approximate $u_{i,j}^{(m)}$ on level 1 (the finest level).

As detailed in Chapters 4 and 5, to formulate the multilevel approach for general level k, we set the following: $b = 2^{k-1}$, $k_1 = (i-1)\,b + 1$, $k_2 = ib$, $\ell_1 = (j-1)\,b + 1$, $\ell_2 = jb$, and $c = (c_{i,j})$. Denoted the current $\tilde{u}$, the minimisation of c in problem (7.35) is equivalent to minimise the following

$$
\begin{aligned}
F_{ES1}\left(c_{i,j}\right) = & \sum_{\ell=\ell_1}^{\ell_2} V_{k_1,\ell}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k_1-1,\ell}-\tilde{u}_{k_1,\ell}\right)\right]^2+\left(\tilde{u}_{k_1-1,\ell}-\tilde{u}_{k_1-1,\ell+1}\right)^2+\beta} \\
& + \sum_{k=k_1}^{k_2-1} V_{k,\ell_2}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k,\ell_2+1}-\tilde{u}_{k,\ell_2}\right)\right]^2+\left(\tilde{u}_{k,\ell_2}-\tilde{u}_{k+1,\ell_2}\right)^2+\beta} \\
& + V_{k_2,\ell_2}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k_2,\ell_2+1}-\tilde{u}_{k_2,\ell_2}\right)\right]^2+\left[c_{i,j}-\left(\tilde{u}_{k_2+1,\ell_2}-\tilde{u}_{k_2,\ell_2}\right)\right]^2+\beta} \\
& + \sum_{\ell=\ell_1}^{\ell_2-1} V_{k_2,\ell}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k_2+1,\ell}-\tilde{u}_{k_2,\ell}\right)\right]^2+\left(\tilde{u}_{k_2,\ell}-\tilde{u}_{k_2,\ell+1}\right)^2+\beta} \\
& + \sum_{k=k_1}^{k_2} V_{k,\ell_1-1}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k,\ell_1-1}-\tilde{u}_{k,\ell_1}\right)\right]^2+\left(\tilde{u}_{k,\ell_1-1}-\tilde{u}_{k+1,\ell_1-1}\right)^2+\beta} \\
& + \frac{1}{2\rho}\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(u_{k,\ell}+c_{i,j}-w_{k,\ell}\right)^2.
\end{aligned}
\tag{7.36}
$$

which can be simplified further using (5.19) as

$$\begin{aligned} F_{ES1}(c_{i,j}) = & \sum_{\ell=\ell_1}^{\ell_2} V_{k_1-1,\ell}\sqrt{(c_{i,j}-h_{k_1-1,\ell})^2+\upsilon_{k_1-1,\ell}^2+\beta} \\ & + \sum_{k=k_1}^{k_2-1} V_{k,\ell_2}\sqrt{(c_{i,j}-\upsilon_{k,\ell_2})^2+h_{k,\ell_2}^2+\beta} \\ & + \sum_{\ell=\ell_1}^{\ell_2-1} V_{k_2,\ell}\sqrt{(c_{i,j}-h_{k_2,\ell})^2+\upsilon_{k_2,\ell}^2+\beta} \\ & + \sum_{k=k_1}^{k_2} V_{k,\ell_1-1}\sqrt{(c_{i,j}-\upsilon_{k,\ell_1-1})^2+h_{k,\ell_1-1}^2+\beta} \\ & + \sqrt{2}V_{k_2,\ell_2}\sqrt{(c_{i,j}-\bar{\upsilon}_{k_2,\ell_2})^2+\bar{h}_{k_2,\ell_2}^2+\frac{\beta}{2}} \\ & + \frac{1}{2\rho}\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}(u_{k,\ell}+c_{i,j}-w_{k,\ell})^2. \end{aligned} \tag{7.37}$$

On the coarsest level $(L+1)$, a **single** constant update for the current $\tilde{u}$ is given as

$$\min_c \{ F_{ES1}(\tilde{u}+c) = \frac{1}{2\rho}\sum_{i=1}^{n}\sum_{j=1}^{n}(u_{i,j}+c-w_{i,j})^2 \} \tag{7.38}$$

which has a simple and explicit solution.

Then, we obtain a multilevel method if we cycle through all levels and all blocks on each level. The overall procedure to solve our first model ES1 is given in Algorithm 8. Again, in order to update the algorithm in a fast manner, we recommend to adjust the fine level before the coarse level and to use the inner iteration $t = 1$.

7.4.3 A multilevel algorithm for ES2

We now consider our model ES2 as expressed by (7.21). The minimiser is obtained by minimisation of $ES2$ with respect to u in (7.26) and w in (7.27) respectively. The solution of (7.27) can be obtained analytically following (7.29). After the solution of u and w are obtained, we update ϕ in (7.29) by reinitilisation procedure solved using Sussman method [131]. Lastly, we update $S(\kappa)$ using (7.31) (equivalent to (7.22)). The iteration is done until convergence achieved. Now, we discuss how to develop a multilevel algorithm to solve (7.26).

The discretised form of the functional $ES2^{2a}(u,w)$ of problem (7.26) is as follows:

$$\begin{aligned} & \min_u ES2^{2a}(u,S,w) \equiv \min_u ES2_\alpha^{2a}(u_{1,1},u_{2,1},...,u_{i-1,j},u_{i,j},u_{i+1,j},...,u_{n,n},S,w) \\ & = \sum_{i=1}^{n}\sum_{j=1}^{n} S_{i,j}\sqrt{(u_{i,j}-u_{i,j+1})^2+(u_{i,j}-u_{i+1,j})^2+\beta} + \frac{1}{2\rho}\sum_{i=1}^{n}\sum_{j=1}^{n}(u_{i,j}-w_{i,j})^2 \end{aligned} \tag{7.39}$$

Consider the local minimisation of (7.39) by the coordinate descent method on the

Algorithm 8 ES1 – Algorithm to solve the new ES1 model

Given image z, an initial guess $u = \tilde{u}$, an initial guess $w = \tilde{w}$, the stop tolerance (tol), and maximum multilevel cycle (maxit) with $L+1$ levels. Compute V, c_1, and c_2.

1) *Solve* (7.17) *to update u using the following steps:*

   i). *Set $u_0 = \tilde{u}$.*

   ii). *Smooth for t iteration the approximation on the finest level 1 that is solve* (7.33) *for $i, j = 1, 2, ...n$*

   iii). *On each coarse level $k = 2, 3, ...L, L+1$:*
   > If $k \leq L$, compute the minimiser c of (7.36)
   > Solve (7.37) *on the coarsest level $k = L+1$*
   > Update $u = \tilde{u} + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b \times b$ block on level k as illustrated in (5.16) of Chapter 5.

2) *Solve* (7.18) *to update w:*

   i). *Set $w_0 = \tilde{w}$.*

   ii). *Compute w using the formula* (7.19).

3) *Update V using* (7.30)

4) *If* $\max\left(\frac{\|u-u_0\|}{\|u_0\|}, \frac{\|w-w_0\|}{\|w_0\|}\right) < tol$ *or maxit is reached, stop the cycle or return to Step 1 with $\tilde{u} = u$ and $\tilde{w} = w$.*

finest level 1:

$$\begin{cases} \text{Given} & u^{(0)} = \left(u_{i,j}^{(0)}\right) \text{ with } m = 0, \\ \text{Solve} & u_{i,j}^{(m+1)} = \underset{u_{i,j} \in \mathbb{R}}{\arg\min}\, ES2_{loc}^{2a}(u, S, w) \text{ for } i, j = 1, 2, ...n, \\ & \text{Repeat the above steps with } m = m+1 \text{ until stopped.} \end{cases} \tag{7.40}$$

where

$$\begin{aligned} ES_{loc}^{2a} &\equiv ES2_{\alpha}^{2a} - ES2_{\bar{\alpha}}^{2a} \\ &= S_{i,j}\sqrt{\left(u_{i,j} - u_{i+1,j}^{(m)}\right)^2 + \left(u_{i,j} - u_{i,j+1}^{(m)}\right)^2 + \beta} \\ &+ S_{i-1,j}\sqrt{\left(u_{i,j} - u_{i-1,j}^{(m)}\right)^2 + \left(u_{i-1,j}^{(m)} - u_{i-1,j+1}^{(m)}\right)^2 + \beta} \\ &+ S_{i,j-1}\sqrt{\left(u_{i,j} - u_{i,j-1}^{(m)}\right)^2 + \left(u_{i,j-1}^{(m)} - u_{i+1,j-1}^{(m)}\right)^2 + \beta} \\ &+ \tfrac{1}{2\rho}(u_{i,j} - w_{i,j})^2. \end{aligned}$$

The terms $ES2_{\bar{\alpha}}^{2a}$ refers to a collection of all terms that are not dependent on $u_{i,j}$. For $u_{i,j}$ at the boundary (the total variation term), Neumann's condition is used. Minimisation of $ES2_{loc}^{2a}$ in (7.40) is achieved using Newton method as similarly done in solving ES1.

To introduce the multilevel algorithm, it is of interest to rewrite (7.40) in an

equivalent form:

$$\begin{cases} \text{Given} & u^{(0)} = \left(u_{i,j}^{(0)}\right) \text{ with } m = 0, \\ \text{Solve} & \hat{c} = \underset{u_{i,j}\in\mathbb{R}}{\arg\min}\, ES2_{loc}^{2a}\left(u_{i,j}^{(m)} + c, S, w\right), \\ \text{Update} & u_{i,j}^{(m+1)} = u_{i,j}^{(m)} + \hat{c}, \\ & \text{Repeat the above steps with } m = m+1 \text{ until stopped.} \end{cases} \tag{7.41}$$

The problem in (7.41) can be interpreted as finding the best correction constant $\hat{c}$ at the current approximate $u_{i,j}^{(m)}$ on level 1 (the finest level).

For general level k, the minimisation of c in problem (7.41) is equivalent to minimise the following

$$\begin{aligned} F_{ES2}\left(c_{i,j}\right) = & \sum_{\ell=\ell_1}^{\ell_2} S_{k_1,\ell}\sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_1-1,\ell} - \tilde{u}_{k_1,\ell}\right)\right]^2 + \left(\tilde{u}_{k_1-1,\ell} - \tilde{u}_{k_1-1,\ell+1}\right)^2 + \beta} \\ & + \sum_{k=k_1}^{k_2-1} S_{k,\ell_2}\sqrt{\left[c_{i,j} - \left(\tilde{u}_{k,\ell_2+1} - \tilde{u}_{k,\ell_2}\right)\right]^2 + \left(\tilde{u}_{k,\ell_2} - \tilde{u}_{k+1,\ell_2}\right)^2 + \beta} \\ & + S_{k_2,\ell_2}\sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_2,\ell_2+1} - \tilde{u}_{k_2,\ell_2}\right)\right]^2 + \left[c_{i,j} - \left(\tilde{u}_{k_2+1,\ell_2} - \tilde{u}_{k_2,\ell_2}\right)\right]^2 + \beta} \\ & + \sum_{\ell=\ell_1}^{\ell_2-1} S_{k_2,\ell}\sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_2+1,\ell} - \tilde{u}_{k_2,\ell}\right)\right]^2 + \left(\tilde{u}_{k_2,\ell} - \tilde{u}_{k_2,\ell+1}\right)^2 + \beta} \\ & + \sum_{k=k_1}^{k_2} S_{k,\ell_1-1}\sqrt{\left[c_{i,j} - \left(\tilde{u}_{k,\ell_1-1} - \tilde{u}_{k,\ell_1}\right)\right]^2 + \left(\tilde{u}_{k,\ell_1-1} - \tilde{u}_{k+1,\ell_1-1}\right)^2 + \beta} \\ & + \frac{1}{2\rho}\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(u_{k,\ell} + c_{i,j} - w_{k,\ell}\right)^2. \end{aligned} \tag{7.42}$$

Using (5.19), the functional (7.42) can be simplified further as

$$\begin{aligned} F_{ES2}\left(c_{i,j}\right) = & \sum_{\ell=\ell_1}^{\ell_2} S_{k_1-1,\ell}\sqrt{\left(c_{i,j} - h_{k_1-1,\ell}\right)^2 + \upsilon_{k_1-1,\ell}^2 + \beta} \\ & + \sum_{k=k_1}^{k_2-1} S_{k,\ell_2}\sqrt{\left(c_{i,j} - \upsilon_{k,\ell_2}\right)^2 + h_{k,\ell_2}^2 + \beta} \\ & + \sum_{\ell=\ell_1}^{\ell_2-1} S_{k_2,\ell}\sqrt{\left(c_{i,j} - h_{k_2,\ell}\right)^2 + \upsilon_{k_2,\ell}^2 + \beta} \\ & + \sum_{k=k_1}^{k_2} S_{k,\ell_1-1}\sqrt{\left(c_{i,j} - \upsilon_{k,\ell_1-1}\right)^2 + h_{k,\ell_1-1}^2 + \beta} \\ & + \sqrt{2}S_{k_2,\ell_2}\sqrt{\left(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2}\right)^2 + \bar{h}_{k_2,\ell_2}^2 + \frac{\beta}{2}} \\ & + \frac{1}{2\rho}\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(u_{k,\ell} + c_{i,j} - w_{k,\ell}\right)^2. \end{aligned} \tag{7.43}$$

On the coarsest level $(L+1)$, a **single** constant update for the current $\tilde{u}$ is given as

$$\min_{c} \{ F_{ES2}(\tilde{u}+c) = \frac{1}{2\rho} \sum_{i=1}^{n} \sum_{j=1}^{n} (u_{i,j} + c - w_{i,j})^2 \} \tag{7.44}$$

which has a simple and explicit solution.

The overall procedure to solve our second model ES2 is given in Algorithm 9.

Algorithm 9 ES2 – Algorithm to solve the new ES2 model

Given image z, an initial guess $u = \tilde{u}$, an initial guess $w = \tilde{w}$, the stop tolerance (tol), and maximum multilevel cycle (maxit) with $L+1$ levels. Compute ϕ, S, c_1, and c_2.

1) *Solve* (7.26) *to update u using the following steps:*

   i). *Set $u_0 = \tilde{u}$.*

   ii). *Smooth for t iteration the approximation on the finest level 1 that is solve* (7.40) *for $i, j = 1, 2, ...n$*

   iii). *On each coarse level $k = 2, 3, ...L, L+1$:*
   > If $k \leq L$, compute the minimiser c of (7.42)
   > Solve (7.43) *on the coarsest level $k = L+1$*
   > Update $u = \tilde{u} + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b \times b$ block on level k as illustrated in (5.16) of Chapter 5.

2) *Solve* (7.27) *to update w:*

   i). *Set $w_0 = \tilde{w}$.*

   ii). *Compute w using the formula* (7.28).

3) *Reinitialise ϕ of* (7.29).

4) *Update S via* (7.31).

5) *If* $\max\left(\frac{\|u-u_0\|}{\|u_0\|}, \frac{\|w-w_0\|}{\|w_0\|}\right) < tol$ *or maxit is reached, stop the cycle or return to Step 1 with $\tilde{u} = u$ and $\tilde{w} = w$.*

7.5 Numerical experiment

In this section, we provide several numerical experiments conducted to show the performance of the proposed algorithms, ES1 and ES2. There are **6 sets** of tests carried out. In the **first set**, we will compare the performance of ES1 and ES2 and chose the best method. The chosen method will be compared with other variational segmentation models. The models are:

Name		Description
SC1	:	A selective segmentation model by Spencer and Chen [129] that is solved using multilevel algorithm developed in Chapter 5 and [74].
SC2	:	A reformulated model of SC1, solved using multilevel algorithm developed in Chapter 5 and reported in [74]. We compare ES with SC1 and SC2 in the **second set** of the tests.
CMT	:	In the **third set**, we compare ES with the latest selective segmentation model by Liu et al [83], called CMT model. We solve the model using multilevel algorithm.
NCZZ	:	A state-of-the-art interactive selective segmentation model by Nguyen et al, called NCZZ [103], solved using Split Bregman method. The comparison of ES with NCZZ is done in the **fourth set** of the tests.
JX	:	A global segmentation model based on Euler's elastica energy by X.C.Tai & J.Duan [133], solved using Split Bregman method. We named the model as JX and compare with ES in the **fifth set** of tests.

In the final test (**test six**), we perform the speed test of the chosen method with an image of different resolutions.

We provide a general guide to choose suitable parameters used in ES1 and ES2 for different images based on our experimental result as follows: parameters $\lambda_1 = \lambda_2 = 1$,$\rho = 10^{-3}$, and $\beta = 10^{-4}$ are fixed throughout the experiments unless stated otherwise. Parameter a ensures the final segmentation curve is smooth and the suitable range for a is $500 \leq a \leq 900$. Parameters b gives more curvature information and connect the broken image boundary. A good range for b is $900 \leq b \leq 1500$. Parameter θ takes large value if the targeted object is too close a nearby object and small θ for well separated object. The standard image size used is 128×128, otherwise if another size is used it will be stated. All algorithms are implemented using MATLAB R2018a on a computer with Intel Core i7 processor, CPU 3.60GHz, 16 GB RAM CPU. The algorithms are stopped when $maxit = 300$ or $tol = 10^{-2}$.

7.5.1 Test Set 1: Comparison of ES1 and ES2

In this test, we consider a synthetic image with different markers set used. The first marker set is shown in Figure 7.1(a) and 7.1(d) shows the second markers set. The final segmentation results by ES1 and ES2 using the markers in 7.1(a) are illustrated in 7.1(b) and 7.1(c) respectively. To complete the segmentation task, ES1 needs 11.5s while ES2 needs 7.6 seconds. The accuracy given by the Jaccard similarity coefficient (JSC) value is the same, that is JSC=0.84. Next, we segment the targeted object using the second markers set in 7.1(d). Figures 7.1(e) and 7.1(f) demonstrate the results by ES1 and ES2 respectively. ES1 needs 23.1 seconds with JSC=0.82 while ES2 needs 7.6 seconds with accuracy, JSC=0.86. Clearly, ES2 is faster than ES1 for both markers set. On top of that, ES2 gives a better final segmentation results compared to ES1. This

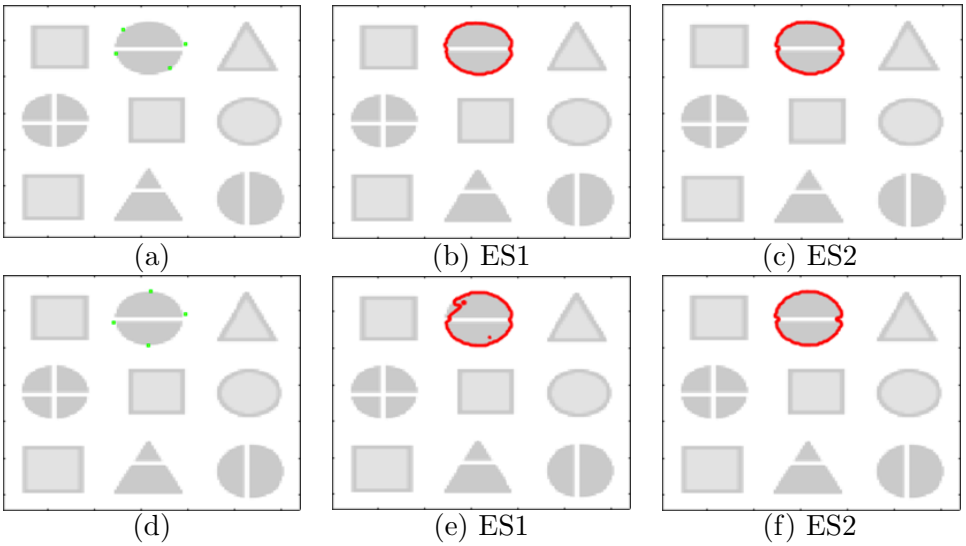


Figure 7.1: Test Set 1—Figure (a) and (d) show the same targeted object with different markers set used. Images (b) and (c) demonstrate the result of ES1 and ES2 respectively using markers in (a). ES1 needs 11.5 seconds while ES2 needs 7.6 seconds with the same accuracy, JSC=0.84. Images (e) and (f) illustrate the result of ES1 and ES2 respectively using markers in (d). ES1 needs 23.1 seconds with JSC=0.82 while ES2 needs 7.6 seconds with accuracy, JSC=0.86. Here, $a = \theta = 500$ and $b = 900$.

implies that ES2 is less dependent to the markers set used. Based on these findings, ES2 is a better algorithm than to ES1. We also plot the initial and final distance function ϕ of equation (7.29) as shown in Figure 7.2 for both marker sets used. The distance function remain a smooth SDF, consequently help in computing an accurate curvature value of a curve. This preservation is important for the success of ES2.

To see how the Euler's elastica affects the final segmentation results of ES2, we fix all other parameters while only varying the curvature parameter b. For this illustration, we use ES2 to segment a real medical image as shown in Figure 7.3(a). The targeted organ occupied with inhomogeneous intensity is marked by the markers. Images 7.3(b) and 7.3(c) show the effect of the curvature parameter $b = 1$ and $b = 900$ respectively. With the aid of the Euler's elastica, one can observed that we able to produce a satisfactory segmentation result for such image for a suitable value of b. As our model formulation is based on the the two-phase (piecewise constant) intensity based segmentation models in [38, 129, 74], notice that, for small value of $b = 1$, all the final segmentation curves inherit the limitation characteristics of the two-phase segmentation models which are unable to restore the broken image boundary and have difficulty in segmenting an image with inhomogeneous of intensity. We show the comparison with other selective segmentation models in the next test set.

7.5.2 Test Set 2: Comparison with SC1 and SC2 models

The aim of this test is to show clearly the comparison of ES2 with other close related two-phase intensity based selective segmentation models namely SC1 and SC2. The

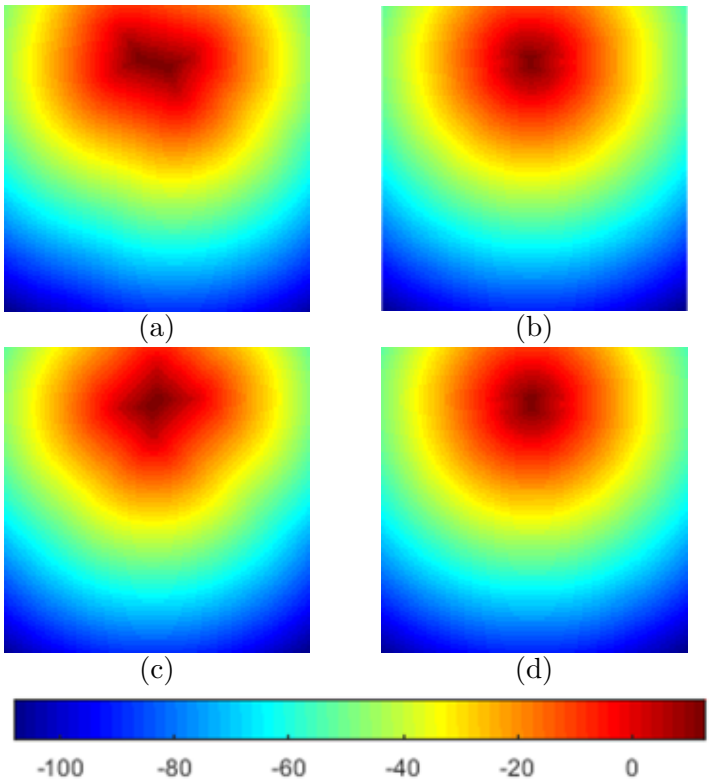


Figure 7.2: Test Set 1—(a) and (c) show the initial distance function ϕ of ES2 in equation (7.29), computed from the corresponding contour generated by the first marker set in 7.1(a) and second markers set in 7.1(d) respectively. Images (b) and (d) illustrate the final distance function ϕ , calculated from the final contour of ES2 in 7.1(c) and 7.1(f) respectively. The color bar indicates the value range of ϕ.

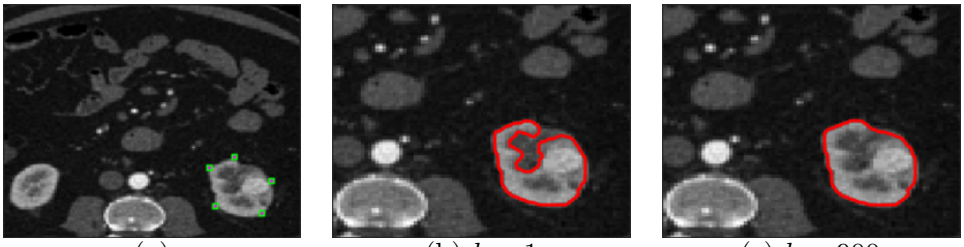


Figure 7.3: Test Set 1— Images (b) and (c) show the effect of the curvature parameter $b = 1$ and $b = 900$ respectively for real medical image in (a) with intensity inhomogenity. Here, ES2 reads $a = 500$ and $\theta = 1500$.

SC1 and SC2 models were reported in the Chapter 5 and in [74]. Figure 7.4(a) and 7.4(b) show the targeted objects with markers used. The image 7.4(a) is the synthetic image with damaged boundary while 7.4(b) is the cap of a mushroom with intensity inhomogenity. Images 7.4(d), 7.4(g), and 7.4(j) demonstrate the segmentation results of the targeted object in 7.4(a) for SC1, SC2, and ES2 respectively, while images 7.4(e), 7.4(h), and 7.4(k) illustrate the segmentation results of the targeted object 7.4(b) for SC1, SC2, and ES2 respectively.

For completeness, we provide in 7.4(c) and 7.4(f) the result of SC1 in segmenting objects in Figure 7.1(a) and 7.3(a) used in test set 1 respectively, while in 7.4(i) and 7.4(l) we demonstrate the results of using SC2 in segmenting objects in Figure 7.1(a) and 7.3(a) of test set 1 respectively. Obviously, SC1 and SC2 are unable to recover the missing boundary of the circle object of images in Figure 7.1(a) and Figure 7.4(a). In addition, SC1 and SC2 also unable to give a good result in segmenting the organ used in Figure 7.3(a) and the cap of the mushroom in 7.4(b) due to intensity inhomogenity of the image. In contrast, the new ES2 which has the additional image curvature information able to fill in the gap of the circle object of image 7.4(a) and give satisfactory result in segmenting the mushroom's cap.

7.5.3 Test Set 3: Comparison with CMT model [83]

In the third experiment, we compare ES2 with the latest selective segmentation model by Liu et al [83], called CMT model. The CMT model is formulated based on the famous Mumford and Shah global segmentation model [102], that assumes piecewise smooth variation of the intensities in an image, suitable for segmentation of images having intensity inhomogeneity. Figure 7.5(a) shows the targeted object with markers used. The final segmentation results by CMT [83] and ES2 are demonstrated in 7.5(b) and 7.5(c) respectively. For completeness, in 7.5(d) we show the result of CMT [83] in segmenting object of Figure 7.3(a) with intensity inhomogenity as used in test set 1. Observed that the CMT model [83] able to deal with image with inhomogenious intensity, the advantage evidence of using piecewise smooth formulation. However it's limitation is that it is unable to segment an object with missing boundary and the model basically needs more number of markers to get a better segmentation results as reported in [74]. For the given synthetic problem in Figure 7.5(a), our ES2 is able to recover the missing boundary.

7.5.4 Test Set 4: Comparison with NCZZ model [103]

In this set, we compare ES2 with the state-of-the-art interactive selective segmentation model by Nguyen et al, called NCZZ [103]. The model is well-known for its robustness in selecting a feature in an image in an interactive way. We acknowledge that the NCZZ model [103] gives satisfactory result in segmenting object with intensity inhomogeneity in Figure 7.3(a) and Figure 7.4(b). For briefness, we will not show the results that both models (NCZZ [103] and ES2) give satisfactory results. This test set will show the performance of NCZZ [103] and ES2 in segmenting object with incomplete boundary.

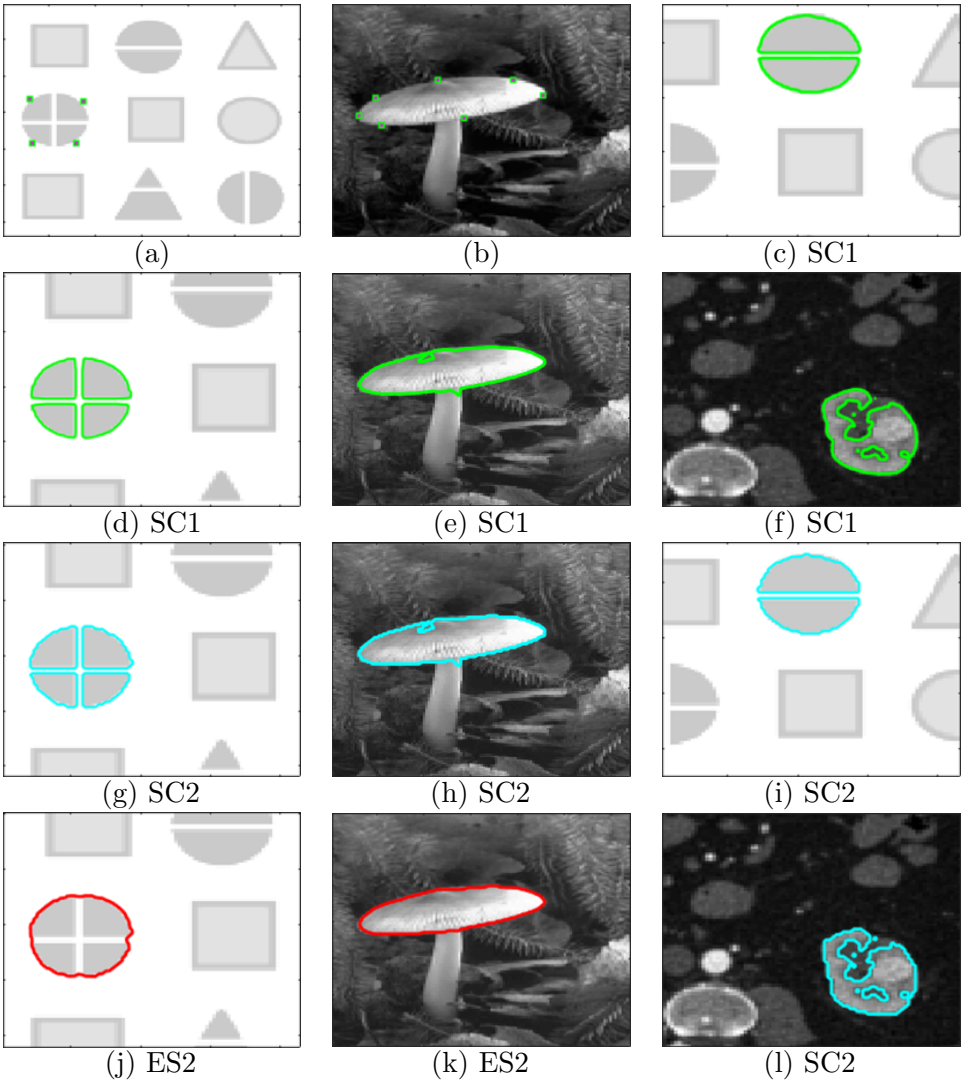


Figure 7.4: Test Set 2—Images (a) and (b) show the targeted objects with markers used. Images (d), (g), and (j) demonstrate the segmentation results of object (a) for SC1, SC2, and ES2 respectively. We used $b = 900$, a and θ equal 500. Images (e), (h), and (k) illustrate the segmentation results of the targeted object (b) that is the cap of the mushroom for SC1, SC2, and ES2 respectively using $\lambda_1 = 0.3$, $a = 900$, $b = 1500$ and $\theta = 3500$. For completeness, in (c) and (f) we show the result of SC1 in segmenting objects in Figure 7.1(a) and 7.3(a) used in test set 1 respectively, while in (i) and (l) show the result of using SC2 in segmenting objects in Fugure 7.1(a) and 7.3(a) of test set 1 respectively.

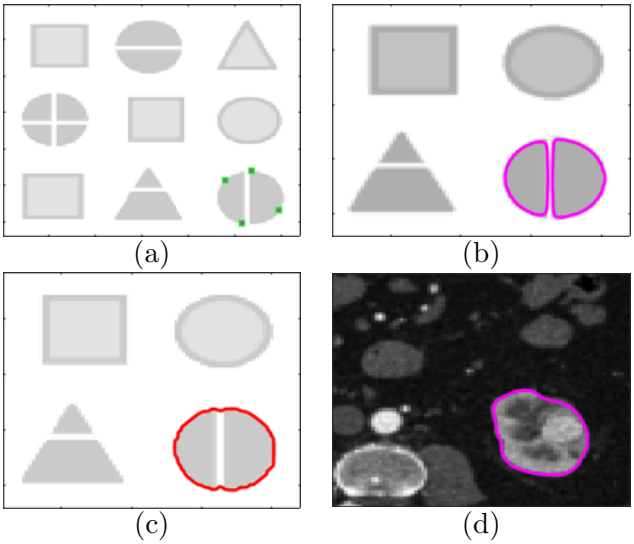


Figure 7.5: Test Set 3—Image (a) shows the targeted object with markers used. Images (b) and (c) demonstrate the segmentation results of CMT [83] and ES respectively. For completeness, in (d) we show the result of CMT [83] in segmenting object of Figure 7.3(a) with intensity inhomogenity as used in test set 1. Here, we used $a = 900$, $b = \theta = 500$.

Figure 7.6(a) and 7.6(b) show the targeted object with markers used for NCZZ model [103] and ES2 respectively. The targeted object is to segment the hat of the kid. However, the hat is partially occluded by the kid. Notice that the NCZZ [103] used both foreground (red) and background (blue) markers while our ES2 just used the forground markers information. The final segmentation result for NCZZ [103] is demonstrated in 7.6(c) while the result of ES2 in binary form is shown in 7.6(d), see also Figure 7.8(b) for the generated segmentation curve by ES2. With only the boundary length regularity information, NCZZ model [103] able to partially segments the hat. In contrast to NCZZ [103], our new model ES2 gives the "prediction" of the occuled part of the hat, closer to the disocclusion of human perception due to the additional curvature regularity information in the ES2 formulation. We remark that the performance of ES2 in segmenting the ellipse shape hat is better than its performance in segmenting ellipse object in Figure 7.1c and 7.1f. This is due to the different position and number of markers set used. Notice that ES2 uses more markers in Figure 7.6b compare to Figure 7.1c and 7.1f. In general, the performance of ES2 is better when more markers are used and the position of markers are placed near or on the targeted object. For a hard problem, for example an object with many incomplete boundaries, it is difficult to determine the position of markers and ES2 may gives unsatisfactory result. To overcome this limitation, one can consider to apply prior shape knowledge idea [32, 43, 46, 81, 128] in the formulation of ES2 and this is worth exploring.

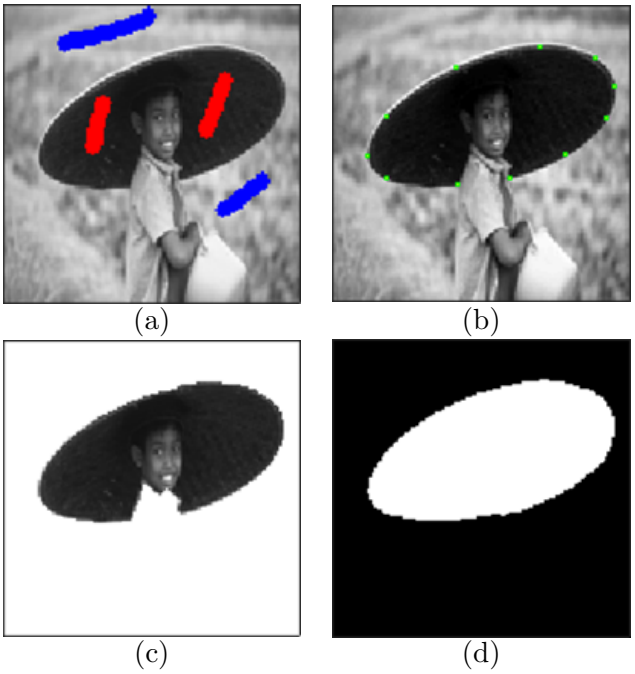


Figure 7.6: Test Set 4—Images (a) and (b) show the targeted object with markers used for NCZZ model [103] and ES2 respectively. Image (c) demonstrate the segmentation result of NCZZ [103] and in (d) we show the result of ES2 respectively. Here, we used $a = 500$, $b = 900$, $\theta = 3000$ and $\lambda_1 = 0.01$.

7.5.5 Test Set 5: Comparison with JX model [132]

We remark that our formulation for selective segmentation model ES2 is based on the formulation of global segmentation model by Tai and Duan [132]. We named the model as JX. Both ES and JX [132] models used Euler's elastica energy. We compare the performance of JX [132] with ES2 in the test set 5 as follows. First, we segment a single targeted object as marked in each Figure 7.1(a), Figure 7.4(a) and Figure 7.4(a) using JX model [132]. The JX model [132] gives the same segmentation result as in 7.7(a) for that segmentation task. This is an expected result because the JX model [132] is not design to select one feature assign by user, except if the targeted object is relatively larger in size compare to other objects appear in a given image. This is a special advantage of JX model [132] where it is able to capture object of large size while "ignoring' the ones of small size [132]. We validate this attribute by a task in segmenting a relatively small incomplete rectangle shape object as marked in 7.7(b). The result is demonstrated in 7.7(c). One can see that the JX model [132] unable to segment the targeted rectangular shape, but it segments a relatively large object that is the incomplete circle object in the image and ignored smaller objects. The satisfactory result delivers by ES2 as shown in 7.7(d). The ES2 also able to segment the largest object (the incomplete circle) in the image. The result is demonstrated in 7.7(f) using markers set in 7.7(e).

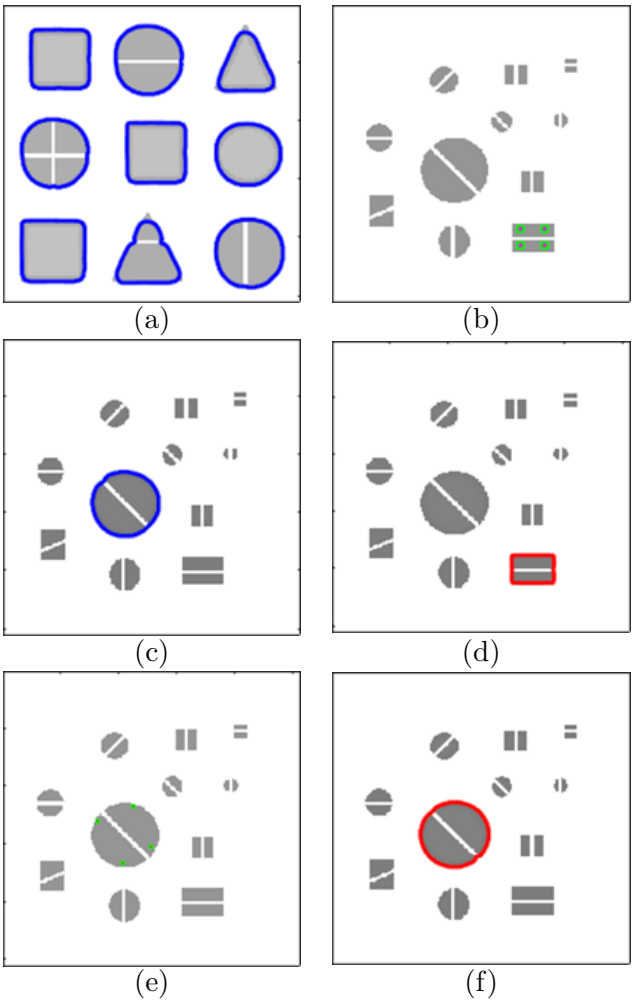


Figure 7.7: Test Set 5—The JX model [132] gives the same segmentation result as in (a) for the task to segment an object in Figure 7.1(a), Figure 7.4(a) and Figure 7.5(a). Image (b) shows the targeted object with marker to be segmented by JX model [132] and ES2. Images (c) and (d) display the result of JX model [132] and ES2 respectively in segmenting image in (b). For completeness of experiment, (f) demonstrates the result of ES2 in segmenting object in (e), the markers are marked by the green boxes. Here, the ES2 reads $a = 900$, $b = 1500$. The parameter $\theta = 500$ for (b) and $\theta = 800$ for (e) with image of size 256×256.

Table 7.1: Test Set 6—CPU time (in seconds) of ES2 in segmenting object in Figure 7.6(b) with different resolutions. The time ratio, close to 4.4 indicates $O(N \log N)$ optimal speed.

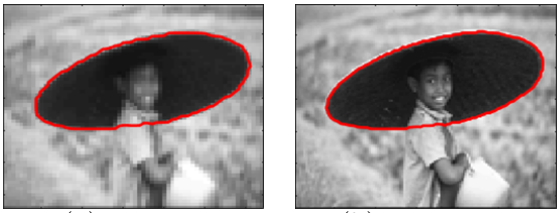
Size	CPU Time	$\frac{t_n}{t_{n-1}}$
64×64	4.3	
128×128	9.2	2.1
256×256	29.8	3.2
512×512	96.7	3.2

7.5.6 Test Set 6: Speed and Convergence of ES2

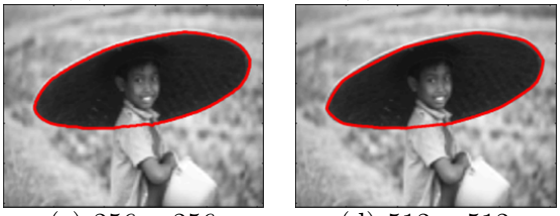
In the final test, we perform the speed test of ES2 in segmenting the targeted object used in Figure 7.6(b) with different image resolutions. All the CPU time (in seconds) are recorded in Table 7.1. Notice that, the ratio of the CPU time in the last column illustrates the optimal complexity of $O(N \log N)$ for multilevel algorithm to solve our model for the image. For the visual quality purpose, the final segmentation curve using image of size 64×64 is demonstrated in Figure 7.8(a) while 7.8(b) is the result for image size of 128×128. Images 7.8(c) and 7.8(d) show the results using image size 256×256 and 512×512 respectively. In addition, to illustrate the convergence of our multilevel algorithm numerically, we give an example of residual curve for ES2 in segmenting image of size 64×64 to reach $tol = eps$ as shown in 7.8(e). Here, eps is relatively small based on a machine epsilon that is, $eps \approx 2.2204 \times 10^{-16}$.

7.6 Summary

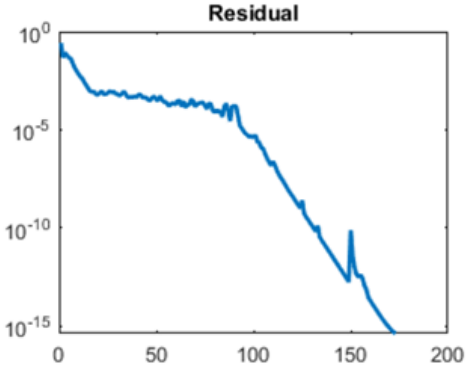
In this chapter, we have presented two Euler's elastica based selective segmentation models called ES1 and ES2. The ES1 model uses the relaxed binary curve representation to compute curvature while ES2 uses the level set of continuous SDF to compute curvature. Comparison carried out between ES1 and ES2 show that ES2 gives more accurate and fast result than ES1. This implies that curve representation using level set of continuous SDF allows to calculate the geometric features of a curve such as normal and curvature more accurate than relaxed binary representation. Consequently, we have chosen ES2 to compare with other related models namely SC1 and SC2. The result indicate that ES2 aided with curvature information is more robust in segmenting object with intensity inhomogenity and abject with incomplete boundary. In the next experiment, we compare ES2 with a new published model in 2018 called CMT by [83]. Result show that both model able to segment object with intensity inhomogenity. However, CMT model [83] unable to restore the missing boundary of targeted object. We also compare ES2 with state-of-the-art interactive selective segmentation model called NCZZ model [103] and the result demonstrated that both models successful in segmenting object with intensity inhomogenity but for object with broken boundary, the NCZZ model [103] unable to give satisfactory result. We also compared ES2 with JX model [132]. In the case where the targeted object is larger than the surrounding object, both JX model [132] and ES2 deliver satisfactory result. When the targeted



(a) 64×64 (b) 128×128



(c) 256×256 (d) 512×512



(e) Residual Curve of (a) to reach $tol = eps$

Figure 7.8: Test Set 6—Final segmentation curve of ES2 in segmenting targeted object in Figure 7.6(b) with different resolution: Images (a) is size of 64×64, (b) 128×128, (c) 256×256, and (d) 512×512. See Table 1 for the CPU time. The residual curve for image size 64×64 to reach $tol = eps$ is shown in (e). Here, eps is relatively small based on a machine epsilon that is, $eps \approx 2.2204 \times 10^{-16}$.

object is smaller than other objects, JX model [132] is unable to segment the object. In final experiment, we demonstrated that the ES2 model solved using developed multilevel algorithm yields optimum speed of $O(N \log N)$.

Chapter 8

Conclusion and Future Work

This thesis presented new algorithms for solving selective segmentation models. There are four classes of variational segmentation models solved using the optimisation based multilevel algorithm; the nonconvex models, convex models, 3-D convex models, and higher order (Euler's Elastica) models.

8.1 Conclusion

In Chapter 4, we first developed a multilevel algorithm for class of nonconvex selective segmentation models; Badshah-Chen [12] and Rada-Chen [111] model with each algorithm reach the expected optimal computation of $O(N \log N)$ complexity for image of size $n \times n = N$. The models were further modified to be the localised version of the original models. This modification allows our multilevel algorithm to compute only within a banded region of an active level set contour and consequently each algorithm having the improve complexity of approximately $O(\sqrt{N} \log N)$.

A second class to consider is convex variational selective segmentation model. There is a way to improve the existence model [129] so that it becomes computationally less expensive and less sensitive to parameters and we discussed the process through primal-dual framework in Chapter 5. We developed multilevel algorithms to solve both the original and the modified model. Experimental results confirmed that the modified model solved using multilevel algorithm is faster and less sensitive to parameters compared to the original model. In order to get a stronger decaying property, a new variant of multilevel algorithm for the modified model was proposed and its convergence was proved. Experiment shows that all the multilevel algorithms reach optimal $O(N \log N)$ complexity.

Next, we developed a new 3-D convex selective segmentation model and 3-D multilevel algorithm in Chapter 6. We also proposed the localised version of the model in order to speed up the computation time. To prove the convergence, we proposed the new variant of multilevel algorithm to solve both the new model and its localised version. Numerical tests show that the proposed models are effective and the multilevel algorithm is efficient in locally segmenting 3-D complex image structures.

Finally, we proposed two new higher order selective segmentation models that apply

Euler's Elastica energy term as boundary regularisation which allow the models to restore those boundaries that are missing or not well defined by the grey intensity images. These models are difficult to solve directly, hence we proposed to treat the models as weighted TV type selective segmentation models and developed multilevel algorithms to solve them. Numerical results demonstrated that the models give satisfactory results to restore the missing object's boundary compared to existing models. Furthermore, the model which used level set information to compute the curvature performed better than the model that used binary relaxation concept in computing the curvature.

8.2 Future Work

There are many different future directions that can be taken from the work presented in this thesis. We mention some of them here:

- All the works in this thesis are based on two-phase selective segmentation problems aided by a foreground marker set . The extension works are to formulate multi-phase selective model with two set of markers that represent foreground and background of the targeted object. In addition, develop multilevel algorithm to solve it would be interesting to explore.
- The Euler's Elastica based selective model gave us a promising result in a 2-D setting. Possible future research is to extend the model in 3-D framework, that could be solved using 3-D multilevel algorithm.
- The development of multilevel algorithm for other imaging problem such as image inpainting and image registration still has not been done and this is worth exploring.
- Another possible future direction is to develop a multilevel algorithm to solve joint selective image segmentation and image registration.
- We will develop other possible smoothers that can be used in the multilevel framework.
- Another interesting future research is to work on modelling selective segmentation model in machine learning framework.

Appendix A

A 3-D Chambolle's projection algorithm for the proposed 3-D SC2 model

Another class of fast algorithm is called Chambolle's projection algorithm [26]. This popular algorithm is considered as a powerful method [33] and a fast method [20, 26, 40]. Here, we propose to extend the Chambolle's projection algorithm [26] into 3-D framework to solve our 3-D selective and convex variational image segmentation problem in (6.10). Again, note that the problem in (6.11) is explicitly solved using 6.12. To develop 3-D Chambolle's method, first define the gradient operator:

$$(\nabla u)_{i,j,k} = \left((\nabla u)^1_{i,j,k}, (\nabla u)^2_{i,j,k}, (\nabla u)^3_{i,j,k} \right)$$

$$(\nabla u)^1_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases}$$

$$(\nabla u)^2_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}$$

$$(\nabla u)^3_{i,j,k} = \begin{cases} u_{i,j,k+1} - u_{i,j,k} & \text{if } k < N \\ 0 & \text{if } k = N. \end{cases}$$

The divergence operator:

$$(\nabla \cdot p)_{i,j,k} = \begin{cases} p^1_{i,j,k} - p^1_{i-1,j,k} & \text{if} \quad 1 < i < N \\ p^1_{i,j,k} & \text{if} \quad i = 1 \\ -p^1_{i-1,j,k} & \text{if} \quad i = N \end{cases} + \begin{cases} p^2_{i,j,k} - p^2_{i,j-1,k} & \text{if} \quad 1 < j < N \\ p^2_{i,j,k} & \text{if} \quad j = 1 \\ -p^2_{i,j-1,k} & \text{if} \quad j = N \end{cases} + \begin{cases} p^3_{i,j,k} - p^3_{i,j,k-1} & \text{if} \quad 1 < k < N \\ p^3_{i,j,k} & \text{if} \quad k = 1 \\ -p^3_{i,j,k-1} & \text{if} \quad k = N. \end{cases}$$

We proceed exactly as in [26, 20]. As shown in [26, 20], equation (6.10) can be written with the dual variable $p = (p_1, p_2)$:

$$\min_u \max_{|p| \leq g} \int_\Omega u \nabla \cdot p \, + \frac{1}{2\rho}(u - w)^2 d\mathbf{x}. \tag{A.1}$$

One can now switch the min and max to obtain the equivalent

$$\max_{|p| \leq g} \min_u \int_\Omega u \nabla \cdot p \, + \frac{1}{2\rho}(u - w)^2 d\mathbf{x}. \tag{A.2}$$

The inner minimization in ((A.2)) is point-wise in u. Carrying it out gives:

$$\nabla \cdot p + \frac{1}{\rho}(u - w) = 0 \Rightarrow u = w - \rho \nabla \cdot p. \tag{A.3}$$

Substituting the expression ((A.3)) for minimal u into the $\max - \min$ problem ((A.2)) gives

$$\max_{|p| \leq g} \int_\Omega (w - \rho \nabla \cdot p) \nabla \cdot p + \frac{\rho}{2}(\nabla \cdot p)^2 d\mathbf{x}. \tag{A.4}$$

Simplifying a bit:

$$\max_{|p| \leq g} \int_\Omega w \nabla \cdot p - \frac{\rho}{2}(\nabla \cdot p)^2 d\mathbf{x}. \tag{A.5}$$

Variation of energy in (A.5) with respect to the vector field p give:

$$\int_\Omega (-\nabla w + \rho \nabla (\nabla \cdot p)) \cdot \delta p \, d\mathbf{x}. \tag{A.6}$$

Along with the point-wise constraint $|p|^2 - g^2 \leq 0$, one gets the optimality condition:

$$-\nabla (\rho \nabla \cdot p - w) + \psi(\mathbf{x}) p = 0. \tag{A.7}$$

Where the Lagrange multiplier $\psi(\mathbf{x}) \geq 0$ for all $\mathbf{x}$. As Chambolle shows in [26], it can be determined and eliminated as follows: if the constraint is not active at a point $\mathbf{x}$,

that is if $|p(\mathbf{x})|^2 < g^2(\mathbf{x})$, then $\psi(\mathbf{x}) = 0$. Otherwise, if the constraint is active at a point $\mathbf{x}$, that is, if $|p(\mathbf{x})|^2 = g^2(\mathbf{x})$, then

$$|\nabla(\rho\nabla \cdot p - w)|^2 - \psi^2 g^2(\mathbf{x}) = 0 \tag{A.8}$$

which leads to the conclusion that in either case, the value of $\psi(\mathbf{x})$ is given by:

$$\psi = \frac{1}{g(\mathbf{x})} |\nabla(\rho\nabla \cdot p - w)|. \tag{A.9}$$

Substituting (A.9) into (A.7) gives:

$$-\nabla(\rho\nabla \cdot p - w) + \frac{1}{g(\mathbf{x})} |\nabla(\rho\nabla \cdot p - w)| p = 0. \tag{A.10}$$

We can use a semi-implicit gradient descent algorithm, as proposed by Chambolle in [26], to solve (A.10) as

$$p^{n+1} = \frac{p^n + \delta t \nabla(\nabla \cdot p^n - w/\rho)}{1 + \frac{\delta t}{g(\mathbf{x})} |\nabla(\nabla \cdot p^n - w/\rho)|}. \tag{A.11}$$

We call the 3-D Chambolle's projection algorithm [26] to solve our 3-D convex and selective segmentation problem as **3DCHB**.

Bibliography

[1] R. Adams and L. Bischof. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(6):641–647, 1994.

[2] A. Ahmad and N. Badshah. Fuzzy selective image segmentation model hybrid with local image data and target region energy. In *Proceedings of 2-Day National Conference on Mathematical Sciences in Engineering Applications (NCMSEA - 2018)*, pages 134–140. University of Engineering and Technology, Peshawar, 2018.

[3] L. Ambrosio and S. Masnou. A direct variational approach to a problem arising in image reconstruction. *Interfaces Free Boundary*, 5(6):63–81, 2003.

[4] L. Ambrosio and S. Masnou. *On a Variational Problem Arising in Image Reconstruction*. Birkhäuser Basel, 2004.

[5] H. Anton. *Elementary Linear Algebra*. Wiley, 1973.

[6] T. Asano, D.Z. Chen, N. Katoh, and T. Tokuyama. Polynomial-time solutions to image segmentation. In *Proceedings of the 7th Ann. SIAM-ACM Conference on Discrete Algorithms*, pages 104–113. SIAM, USA, 1996.

[7] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer-Verlag, New York, USA, 2001.

[8] J. F. Aujol and A. Chambolle. Dual norms and image decomposition models. *Int. J. of Comp. Vision*, 63(1):86–104, 2005.

[9] J. F. Aujol, G. Gilboa, T. F. Chan, and S. Osher. Structure-texture image decomposition-modelling, algorithms, and parameter selection. *Int. J. of Comp. Vision*, 67(1):111–136, 2006.

[10] N. Badshah and K. Chen. Multigrid method for the chan-vese model in variational segmentation. *Comm. in Comp. Physics*, 4(2):294–316, 2008.

[11] N. Badshah and K. Chen. On two multigrid algorithms for modelling variational multiphase image segmentation. *IEEE Trans. on Image Processing*, 18(5):1097–1106, 2009.

[12] N. Badshah and K. Chen. Image selective segmentation under geometrical constraints using an active contour approach. *Comm. in Comp. Physics*, 7(4):759–778, 2010.

[13] E. Bae, J. Shi, and X.C. Tai. Graph cuts for curvature based image denoising. *IEEE Trans. Image Processing*, 20(1):1199–1210, 2011.

[14] E. Bae, X.C. Tai, and W. Zhu. Augmented lagrangian method for an euler's elastica based segmentation model that promotes convex contours. *Inv. Prob. and Imaging*, 11(1):1–23, 2017.

[15] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *IEEE ICCV*, pages 1–8. IEEE, 2007.

[16] I. N. Bankman, I. Simon T. Nizialek, O. B. Gatewood, I. N. Weinberg, and W. R. Brody. Segmentation algorithms for detecting microcalcifications in mammograms. *IEEE Trans. Inform. Techn. Biomed.*, 1(2):161–173, 1993.

[17] A. L. Bertozzi and J. B. Greer. Low curvature image simplifiers: global regularity of smooth solutions and laplacian limiting schemes. *Comm. Pure Appl. Math.*, 57(2):764–790, 2004.

[18] S. Beucher. Segmentation tools in mathematical morphology. In *Proceedings of the SPIE on Image Algebra and Morphological Image Processing, 1350*, pages 70–84. SPIE, 1990.

[19] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31:333–390, 1977.

[20] X. Bresson, S. Esedoglu, P. Vandergheynst, J. P. thiran, and S. Osher. Fast global minimization of the active contour/snake model. *J. of Math. Imaging Vision*, 28:151–167, 2007.

[21] C. Brito-Loeza and K. Chen. Fast numerical algorithms for euler's elastica inpainting model. *Int. J. of Modern Mathematics*, 5:157–182, 2010.

[22] C. Brito-Loeza, K. Chen, and V. Uc-Cetina. Image denoising using the gaussian curvature of the image surface. *Numer. Methods Partial Differential Eq.*, 32:1066–1089, 2016.

[23] X. Cai, R. Chan, and T. Zeng. A two-stage image segmentation method using a convex variant of the mumford-shah model and thresholding. *SIAM J. of Imaging Sciences*, 6:368–390, 2013.

[24] J. L. Carter. *Dual Method for Total Variation-Based Image Restoration.* PhD thesis, Department of Mathematics, University of California, Los Angeles, CA, USA, 2002.

[25] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. of Computer Vision*, 22(1):61–79, 1997.

[26] A. Chambolle. An algorithm for total variation minimization and applications. *J. of Math. Imaging and Vision*, 20(1-2):89–97, 2004.

[27] A. Chambolle and P. L. Lions. Image recovery via total variation minimization and related problems. *Numer. Math*, 76(1-2):167–188, 1997.

[28] A. Chambolle and T. Pock. A first order primal-dual algorithm for convex problems with applications to imaging. *J. of Math.l Imaging and Vision*, 40:120–145, 1997.

[29] R. H. Chan and K. Chen. Multilevel algorithm for a poisson noise removal model with total variation regularisation. *Int. J. Comput. Math.*, 84(8):1183–1198, 2007.

[30] R. H. Chan and K. Chen. A multilevel algorithm for simultaneously denoising and deblurring images. *SIAM J. Sci. Comput.*, 32(2):1043–1063, 2010.

[31] T. Chan and J. Shen. *Image Processing and Analysis*. SIAM, 2005.

[32] T. Chan and W. Zhu. Level set based shape prior segmentation. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1164–1179. IEEE, 2005.

[33] T. F. Chan and K. Chen. An optimization-based multilevel algorithm for total variation image denoising. *Multiscale Model. Simul.*, 5(2):615–645, 2006.

[34] T.F. Chan, S. Esedoglu, and M. Nikilova. Algorithm for finding global minimizers of image segmentation and denoising models. *SIAM J. on Applied Mathematics*, 66(5):1632–1648, 2006.

[35] T.F. Chan, G. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. on Scientific Computing*, 20(6):1964–1977, 1999.

[36] T.F. Chan, S. H. Kang, and J. H. Shen. Euler's elastica and curvature based inpaintings. *SIAM J. Appl. Math*, 63(2):564–592, 2002.

[37] T.F. Chan, B. Y. Sandberg, and L. A. Vese. Active contours without edges for vector-valued images. *J. Vis. Commun. Image Represent*, 11(2):130–141, 2000.

[38] T.F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, 2001.

[39] T.F. Chan and L. A. Vese. *Active Contour and Segmentation Models using Geometric PDE's for Medical Imaging. In: Malladi R. (eds) Geometric Methods in Bio-Medical Image Processing. Mathematics and Visualization*. Springer, Berlin, Heidelberg, 2002.

[40] D. Chen, M. Yang, and L. D. Cohen. Global minimum for a variant mumford-shah model with application to medical image segmentation. *Comp. Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, 1(1):48–60, 2013.

[41] L. Chen, Y. Zhou, Y. Wang, and J. Yang. Gacv: Geodesic-aided c–v method. *Pattern Recognition*, 39(7):1391–1395, 2006.

[42] Y. Chen, S. Levine, and M. Rao. Variable exponent, linear growth functionals in image restoration. *SIAM J.l on Applied Mathematics*, 66(4):1383–1406, 2006.

[43] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. Gopinath, R. Briggs, and E. Geiser. Using prior shapes in geometric active contours in a variational framework. *Int. J. of Computer Vision*, 50(1):315–328, 2002.

[44] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[45] M. Comer, C. Bouman, and J. Simmons. Statistical methods for image segmentation and tomography reconstruction. *Microscopy and Microanalysis*, 16(S2):1852–1853, 2010.

[46] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *Int. J. of Computer Vision*, 50(1):295–313, 2002.

[47] T. Dietenbeck, M. Alessandrini, D. Friboulet, and O. Bernard. Creaseg: a free software for the evaluation of image segmentation algorithms based on level-set. In *IEEE International Conference On Image Processing, Hong Kong, China*, pages 665–668. IEEE, 2010.

[48] J. Duan, Z. Pan, B. Zhang, W. Liu, and X. C. Tai. Fast algorithm for color texture image inpainting using the non-local ctv model. *J. of Global Optimization*, 62(4):853–876, 2015.

[49] J. Duan, Z. Qiu, W. Lu, G. Wang, Z. Pan, and L. Bai. An edge weighted second order variational model for image decomposition. *Dig. Signal Processing*, 49(4):162–181, 2016.

[50] 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. *Phy. in Medicine and Biology*, 60(22):8901–8922, 2015.

[51] N. Y. El-Zehiry and L. Grady. Fast global optimization of curvature. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and, Pattern Recognition*, pages 3257–3264. IEEE, 2010.

[52] M. Elsey and S. Esedoglu. Analogue of the total variation denoising model in the context of geometry processing. *SIAM J. Multiscale Model. Simul.*, 7:1549–1573, 2009.

[53] S. Esedoglu and R. March. Segmentation with depth but without detecting junctions. *J. of Math. Imaging and Vision*, 18:7–15, 2003.

[54] S. Esedoglu and J. Shen. Digital inpainting based on the mumford-shah-euler image model. *Eur. J. Appl. Math.*, 13:353–370, 2002.

[55] H. Federer. *Geometric Measure Theory*. Springer, 1969.

[56] W. Fleming and R. Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11(1):218–222, 1960.

[57] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice-Hall, Inc, 1963.

[58] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(6):721–741, 1984.

[59] P. Getreuer. Rudin-osher-fatemi total variation denoising using split bregman. *Image Processing On Line*, 2:74–95, 2012.

[60] M. Giaquinta and S. Hildebrandt. *Minimal Surfaces and Functions of Bounded Variation, volume 80 of Monographs in Mathematics*. Birkhauser Boston, 1984.

[61] M. Giaquinta and S. Hildebrandt. *Calculus of Variations I, The Lagrangian Formalism*. Springer-Verlag, 1996.

[62] E. Giusti. *Calculus of Variations II, The Hamiltonian Formalism*. Springer-Verlag, 1996.

[63] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM J. on imaging sciences*, 2(2):323–343, 2009.

[64] C. Gout and C. Le Guyader. Segmentation of complex geophysical structures with well data. *Comp.l Geosciences*, 10(4):361–372, 2006.

[65] C. Gout, C. Le Guyader, and L.A. Vese. Segmentation under geometrical conditions with geodesic active contour and interpolation using level set methods. *Num.l Algorithms*, 39:155–173, 2005.

[66] L. Grady. Random walks for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[67] C.L. Guyader and C. Gout. Geodesic active contour under geometrical conditions theory and 3d applications. *Num. Algorithms*, 48:105–133, 2008.

[68] J. Hadamard. Sur les problemes aux derivees partielles et leur signi
cation physique. *Princeton University Bulletin*, pages 49–52, 1902.

[69] M. Hadhoud, M. Amin, and W. Dabbour. Detection of breast cancer tumor algorithm using mathematical morphology and wavelet analysis. In *Proceedings of GVIP 05 Conference, CICC, Cairo, Egypt*, pages 75–80. ICGST, 2005.

[70] X. He, W. Zhu, and X. C. Tai. Segmentation by elastica energy with l^1 and l^2 curvatures: a performance comparison. *Num. Math. : Theory, Methods and Applications*, 12(1):285–311, 2019.

[71] M. Ibrahim. *Variational Models and Numerical Algorithms for Effective Image Registration.* PhD thesis, University of Liverpool, 2015.

[72] N. Jamil and K. Awang. *Practical Digital Image Processing with MATLAB.* UiTM Press, Malaysia, 2014.

[73] A. K. Jumaat and K. Chen. An optimization-based multilevel algorithm for variational image segmentation models. *Elect. Trans. on Numerical Analysis*, 46:474–504, 2017.

[74] A. K. Jumaat and K. Chen. A reformulated convex and selective variational image segmentation model and its fast multilevel algorithm. *Num. Math. : Theory, Methods and Applications*, 12:403–437, 2018.

[75] S. H. Kang, W. Zhu, and J. Shen. Illusory shapes via corner fusion. *SIAM J. on Imaging Sciences*, 7:1907–1936, 2014.

[76] G. Kanizsa. *Organization in Vision: Essays on Gestalt Perception.* Praeger, New York, USA, 1979.

[77] M. Kass, M. Witkin, and D. Terzopoulos. Snakes: active contour models. *Inter. J. Comput. Vision*, 1:321–331, 1987.

[78] A. Kenigsberg, R. Kimmel, and I. Yavneh. A multigrid approach for fast geodesic active contours. Tech. rep. cis-2004-06, Computer Science Department, The Technion-Israel Int. Technol., Haifa, 2004.

[79] L. P. Lebedev and I. I. Vorovich. *Functional Analysis in Mechanics.* Springer Verlag, 2002.

[80] D. Lesnic. Characterizations of the functions with bounded variations. *Acta Universitatis Apulensis,*, 6:47–54, 2003.

[81] M. Leventon, W. Grimson, and O. Faugeraus. Statistical shape influence in geodesic active contours. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 316–323. IEEE, 2000.

[82] C. Li, C. Kao, J. C. Gore, and Z. Ding. Implicit active contours driven by local binary fitting energy. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2007.

[83] C. Liu, M. K. P. Ng, and T. Zeng. Weighted variational model for selective image segmentation with application to medical images. *Pattern Recognition*, 16:367–379, 2018.

[84] A. E. H. Love. *A Treatise on the Mathematical Theory of Elasticity, 4th ed.* Dover, New York, 1927.

[85] T. Lu, P. Neittaanmaki, and X. C. Tai. A parallel splitting-up method for partial differential equations and its application to navier-stokes equations. *RAIRO: Mathematical Modelling and Numerical Analysis,*, 26:673–708, 1992.

[86] M. Lysaker, A. Lundervold, and X.C. Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Trans. on Image Processing*, 12:1579–1590, 2003.

[87] M. Lysaker, S. Osher, and X.C. Tai. Noise removal using smoothed normals and surface fitting. *IEEE Trans. on Image Processing*, 13:1345–1357, 2004.

[88] J. Malik, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. J. of Computer Vision*, 43:7–27, 2001.

[89] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:158–175, 1995.

[90] S. Mallat. *A Wavelet Tour Of Signal Processing.* Academic Press, USA, 1998.

[91] A. Marquina and S. Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM J. on Scientific Computing*, 22:387–405, 2000.

[92] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int. Conf. Computer Vision*, pages 416–423, 2001.

[93] S. Masnou. Disocclusion: a variational approach using level lines. *IEEE Trans. Image Processing*, 11:68–76, 2002.

[94] S. Masnou and J. M. Morel. Level lines based disocclusion. In *Proceedings IEEE International Conference on Image Processing*, pages 259–263, 1998.

[95] S. Masnou and J. M. Morel. On a variational theory of image amodal completion. *Rend. Sem. Mat. Univ. Padova*, 116:211–252, 2006.

[96] J. Mille, R. Bone, P.Makris, and H. Cardot. Narrow band region-based active contours and surfaces for 2d and 3d segmentation. *Computer Vision and Image Understanding*, 113:946–965, 2009.

[97] L. Moisan. How to discretize the total variation of an image? In *Procedings of Appl. Math. Mech., John Wiley and Sons, Ltd*, pages 1041907–1041908, 2007.

[98] J.M. Morel and S. Solimini. *Variational methods in image segmentation.* Birkhauser Boston Inc., Cambridge, MA, USA, 1995.

[99] J. C. Moreno, V. B. S. Prasath, H. Proenca, and K. Palaniappan. Fast and globally convex multiphase active contours for brain mri. *Comp. Vision and Image Understanding*, 125:237–250, 2014.

[100] S. Morigi, L. Reichel, and F. Sgallari. Noise-reducing cascadic multilevel methods for linear discrete ill-posed problems. *Num. Algorithms*, 53:1–22, 2010.

[101] D. Mumford. *Elastica and computer vision, , in Algebraic Geometry and Its Applications.* Springer-Verlag, New York, 1994.

[102] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Comm. on Pure Applied Mathematics*, 42:577–685, 1989.

[103] T. N. A. Nguyen, J. Cai, J. Zhang, and J. Zheng. Robust interactive image segmentation using convex active contours. *IEEE Trans. on Image Processing*, 21(8):3734–3743, 2012.

[104] M. Nitzberg, D. Mumford, and T. Shiota. *Filering, Segmentation, and Depth.* Springer-Verlag Berlin Heidelberg, 1993.

[105] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, London, 1970.

[106] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 2003.

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

[108] G. Papandreou and P. Maragos. A fast multigrid implicit algorithm for the evolution of geodesic active contours. In *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition, Washington DC*, pages 689–694. IEEE, 2004.

[109] G. Papandreou and P. Maragos. Multigrid geometric active contour models. *IEEE Trans. Image Process.*, 16:229–240, 2007.

[110] L. Rada and K. Chen. A new variational model with dual level set functions for selective segmentation. *Comm. in Comp. Physics*, 12(1):261–283, 2012.

[111] L. Rada and K. Chen. Improved selective segmentation model using one level set. *J. of Algo. and Computational Technology*, 7:509–541, 2013.

[112] L. Rada and K. Chen. On a variational model for selective image segmentation of features with infinite perimeter. *J. of Math. Research with Applications*, 33(3):253–272, 2013.

[113] L. Rada and K. Chen. A variational model and its numerical solution for local, selective and automatic segmentation. *Num. Algorithm*, 66:399–430, 2014.

[114] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *J. ACM Trans. on Graphics*, 23(3):309–314, 2004.

[115] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[116] J. Savage and K. Chen. An improved and accelerated non-linear multigrid method for total-variation denoising. *Int. J. of Computer Mathematics*, 82:1001–1015, 2005.

[117] J. Savage and K. Chen. On multigrids for solving a class of improved total variation based staircasing reduction models. *Img. Proces. Based On Partial Differential Equations, eds. X.-C. Tai, K.-A. Lie, T. F. Chan and S. Osher*, 82:69–94, 2006.

[118] M. Schechter. *Principles of Functional Analysis*. AMS, 2001.

[119] T. Schoenemann, F. Kahl, and D. Cremers. Curvature regularity for region-based image segmentation and inpainting: a linear programming relaxation. In *IEEE Int. Conf. Comput. Vission*, pages 17–23. IEEE, 2009.

[120] T. Schoenemann, F. Kahl, S. Masnou, and D. Cremers. A linear framework for region-based image segmentation and inpainting involving curvature penalization. *Int. J. of Computer Vision*, 99:53–68, 2012.

[121] D. Sen and S. K. Pal. Histogram thresholding using fuzzy and rough measures of association error. *IEEE Trans. on Image Processing*, 18:879–888, 2009.

[122] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proceedings of the National Academy of Sciences*, pages 1591–1595. National Academy of Sciences, 1996.

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

[124] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, Cambridge, UK, 2002.

[125] J. A. Sethian and R. Fedkiw. Level set methods: An overview and some recent results. *J. of Comp. Physics*, 169:463–502, 2001.

[126] A. I. Shihab. *Fuzzy Clustering Algorithms and Their Application to Medical Image Analysis*. PhD thesis, Dept. of Computing, Univ. of London, 2000.

[127] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis. and Machine Vision*. Chapman and Hall, 1998.

[128] J. Spencer. *Variational methods for image segmentation*. PhD thesis, University of Liverpool, 2016.

[129] J. Spencer and K. Chen. A convex and selective variational model for image segmentation. *Comm. in Math. Sciences*, 13:1453–1452, 2015.

[130] G. Strang. Maximal flow through a domain. *Math. Programming*, 26:123–143, 1983.

[131] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. of Comp. Physics*, 114:146–159, 1994.

[132] X. C. Tai and J. Duan. A simple and fast algorithm for minimization of the elastica energy combining binary and level set representation. *Int. J. of Numerical Analysis and Modelling*, 14:809–821, 2017.

[133] X. C. Tai, J. Hahn, and G. J. Chung. A fast algorithm for euler's elastica model using augmented lagrangian method. *SIAM J. Imaging Science*, 4:313–344, 2011.

[134] L. Tan, Z. Pan, W. Liu, J. Duan, W. Wei, and G. Wang. Image segmentation with depth information via simplified variational level set formulation. *J. of Math. Imaging and Vision*, 60:1–17, 2017.

[135] A. N. Tikhonov and V.Y. Arsenin. *Solutions of ill-posed problems*. New York: Winston,, 1977.

[136] Y.H.R. Tsai, L.T. Cheng, S. Osher, and H.K. Zhao. Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM J. on Numerical Analysis*, 41(2):673–694, 2003.

[137] S. E. Umbaugh. *Computer Vision and Image Processing: A Practical Approach Using CVIPTools*. Prentice Hall PTR, US, 1997.

[138] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. Comput. Vision*, 50:271–293, 2002.

[139] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. on Scientific Computing*, 17:227–238, 1996.

[140] J. Weickert. *Anisotropic Diffusion in Image Processing.* B.G. Teubner, 1998.

[141] J. Weickert, B. M. ter Haar Romeny, and Max A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. on Image Processing*, 7:398–410, 1998.

[142] J. Weickertn. From the continuous to the discrete setting. In *12th International Conference on Analysis and Optimization of Systems, Images, Wavelets and PDEs of Lecture Notes in Control and Information Sciences*, pages 111–118. Springer, 1996.

[143] J. S. Weszka. A survey of threshold selection techniques. *Comp. Graph. and Image Proc.*, 7:259–265, 1978.

[144] L. W. Wilson. Unconditionally stable time splitting methods for the electrostatic analysis of solvated biomolecules. Master's thesis, The University of Alabama, 2015.

[145] C. Wu and X. C. Tai. Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models. *SIAM J. Imaging Science*, 3:300–339, 2010.

[146] X. Xie and M. Mirmehdi. Radial basis function based level set interpolation and evolution for deformable modelling. *Imag. Vision Computing*, 29:167–177, 2011.

[147] M. Yashtini and S.H. Kang. *Alternating Direction Method of Multiplier for Euler's Elastica-Based Denoising. In: Aujol JF., Nikolova M., Papadakis N. (eds) Scale Space and Variational Methods in Computer Vision, SSVM 2015, Lecture Notes in Computer Science.* Springer, Cham, 2015.

[148] M. Yashtini and S.H. Kang. A fast relaxed rormal two split method and an efective weighted tv approach for euler's elastica image inpainting. *SIAM J. Imaging Sciences*, 9:1552–1581, 2016.

[149] Y.-L. You and M. Kaveh. Fourth-order partial differential equations for noise removal. *IEEE Trans. on Image Processing*, 9:1723–1730, 2000.

[150] J. Zhang, K. Chen, and D. Gould. A fast algorithm for automatic segmentation and extraction of a single object by active surfaces. *Int. J. of Comp. Mathematics*, 92:1251–1274, 2015.

[151] J. Zhang, K.Chen, and B.Yu. A multigrid algorithm for the 3d chan-vese model of variational image segmentation. *Int. J. of Comp. Mathematics*, 89:160–189, 2011.

[152] J. Zhang, K.Chen, B.Yu, and D.Gould. A local information based variational model for selective image segmentation. *J. Inv. Problems and Imaging*, 8:293–320, 2014.

[153] H. Zhao. A fast sweeping method for eikonal equations. *Math. of Computation*, 74(250):603–627, 2005.

[154] W. Zhu and T. Chan. A variational model for capturing illusory contours using curvature. *J. of Math. Imaging and Vision*, 27:29–40, 2007.

[155] W. Zhu, T. Chan, and S. Esedoglu. Segmentation with depth: A level set approach. *SIAM J. on Sci. Computing*, 28:1957–1973, 2006.

[156] W. Zhu, X. C. Tai, and T. Chan. Image segmentation using euler's elastica as the regularization. *J. of Sci. Computing*, 57:414–438, 2013.