

Variational Methods for Image Segmentation

by

Jack A. SPENCER

under the supervision of:

Prof. Ke CHEN



UNIVERSITY OF
LIVERPOOL

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

July 2016

Contents

Acknowledgements	v
Abstract	vi
Publications and Presentations	viii
List of Acronyms	ix
List of Figures	x
List of Tables	xviii
1 Introduction	1
1.1 Image Segmentation	1
1.2 Thesis Outline	4
1.3 Contribution	6
2 Mathematical Preliminaries	8
2.1 Linear Vector Spaces	8
2.1.1 Normed Linear Spaces	9
2.1.2 Convex Sets and Functions	11
2.2 Functions of Bounded Variation	12
2.2.1 Co-area Formula	14
2.2.2 Derivative of a BV Function	15
2.3 Inverse Problems	15
2.3.1 Well and Ill-Posed Problems	15
2.3.2 Regularisation	16
2.4 Calculus of Variations	16
2.4.1 Variation of a Functional	17
2.4.2 Gâteaux Derivative of a Functional	17
2.4.3 The Divergence Theorem	18
2.5 Discretisation of Partial Differential Equations	19
2.6 Interface Representation	21
2.6.1 Curves in Euclidean Spaces	21
2.6.2 Heaviside and Dirac delta function	21
2.6.3 Level Set Method	22
2.7 Iterative Solutions to Equations	24

2.7.1	Basic Methods for Linear Systems	24
2.7.2	Nonlinear Equations	27
3	Review of Variational Methods for Imaging Processing	33
3.1	Introduction	33
3.1.1	Denoising	34
3.1.2	Deblurring	36
3.1.3	Registration	38
3.2	Image Segmentation	40
3.2.1	Mumford-Shah Approach	41
3.2.2	Geodesic Active Contours	42
3.2.3	Active Contours Without Edges	43
3.2.4	Convex Relaxation Methods	46
3.3	Algorithms with Applications to Imaging	48
3.3.1	Chambolle's Dual Formulation	49
4	Additive Operator Splitting for Globally Convex Segmentation	51
4.1	Introduction	51
4.2	Globally Convex Segmentation	52
4.2.1	Gradient Descent	53
4.2.2	Dual Formulation	54
4.3	Finding the Global Minimum	55
4.3.1	Introducing a New Regularised Penalty Function	55
4.3.2	Convexity of the Proposed Functional	57
4.3.3	Deriving the Euler-Lagrange Equation	58
4.4	A New Additive Operator Splitting Scheme for GCS	60
4.4.1	Method 1	60
4.4.2	Method 2	61
4.5	Experimental Results	62
4.5.1	Test Set 1 (AOS Parameters)	64
4.5.2	Test Set 2 (Dual Formulation Comparison)	65
4.5.3	Test Set 3 (Initialisation Dependence)	67
4.6	Remarks	77
5	Global Minimisers of Selective Segmentation Models	79
5.1	Introduction	79
5.2	Global Segmentation	80
5.2.1	Piecewise-Constant Mumford-Shah	81
5.2.2	Two-Phase Chan-Vese	81
5.2.3	A Global Convex Reformulation	83
5.3	The Selective Segmentation Problem and Recent Models	83
5.4	Proposed Distance Selective Segmentation Model	86
5.4.1	A New Nonconvex Selective Model	86
5.4.2	A Selective Convex Reformulation	87

5.4.3	Unconstrained Minimisation	88
5.4.4	Numerical Implementation	89
5.5	Experimental Results	91
5.5.1	Test Set 1 (Nonconvex Model Comparisons)	92
5.5.2	Test Set 2 (Robustness to User Selection)	93
5.5.3	Test Set 3 (Improved AOS Method)	94
5.5.4	Test Set 4 (Medical Applications)	95
5.6	Remarks	95
6	Segmentation with Intensity Inhomogeneity	101
6.1	Introduction	101
6.2	Variant Mumford-Shah Model	103
6.2.1	Convergence Behaviour of VMS	103
6.3	Stabilised Bias Field	104
6.3.1	Relationship to Chan-Vese and Mumford-Shah	107
6.3.2	Iterative Minimisation of SBF Formulation	107
6.3.3	Numerical Implementation	108
6.3.4	Experimental Results	110
6.4	Selective Segmentation with SBF	114
6.4.1	Experimental Results	116
6.5	Remarks	117
7	Simultaneous Reconstruction and Segmentation	119
7.1	Introduction	119
7.2	Existing Methods	120
7.2.1	Segmentation of Blurred Images	120
7.2.2	Two-Stage Approach for Images with Known Blur	121
7.3	Segmentation of Images Corrupted By Unknown Blur	122
7.3.1	Two-Stage Approach	122
7.3.2	A Joint Model for Blind Image Segmentation	126
7.4	A Relaxed Model for Blind Image Segmentation	129
7.5	Experimental Results	130
7.5.1	Test Set 1 (Two-Stage Comparisons)	132
7.5.2	Test Set 2 (Significant Blur)	133
7.5.3	Test Set 3 (Joint Model Comparisons)	133
7.6	Remarks	134
8	Incorporating Shape Priors in Variational Segmentation	143
8.1	Introduction	143
8.2	Background and Related Models	144
8.2.1	Level Set Based Shape Prior Segmentation	145
8.2.2	Interactive Shape Prior Segmentation	145
8.2.3	Shape Representation	146
8.3	Motivation	146

8.4	Proposed Two-Stage Shape Prior Model	149
8.4.1	Stage 1: Affine Registration	150
8.4.2	Stage 2: Segmentation	152
8.4.3	Two-Stage Algorithm	154
8.5	Experimental Results	154
8.5.1	Test Set 1 (Occlusions)	155
8.5.2	Test Set 2 (Parameter Dependence)	155
8.5.3	Test Set 3 (Sequential Selection)	160
8.6	Remarks	162
9	Conclusions and Future Work	168

Acknowledgements

It is important to thank the department for this opportunity and their financial funding, as well as their patience during my undergraduate studies. I would also like to register my gratitude with those in the department that have helped me throughout my eight years at the University of Liverpool. In particular, the support of John Gracey, Mary Rees, and Özgür Selsil was integral to me becoming a PhD student, and is greatly appreciated. Over the past four years the advice of Alexander Movchan and Martyn Hughes, among many others, has been invaluable. There are too many to mention here in full, but I have always welcomed their consistent encouragement. I would also like to thank my external examiner, Tim Cootes, for making my viva voce a pleasant experience, and for his comments and advice.

I should acknowledge everyone involved with CMIT during my time here, including visitors and fellow students. The guidance of Lavdie Rada in the early stages of my research was vital, and the feedback of Bryan Williams before my thesis submission was particularly helpful. I must also mention Mazlinda Ibrahim, Jianping Zhang, and Tuomo Valkonen for their useful contributions as part of the group; my understanding of this work is certainly greater because of them.

Finally, I would like to thank Ke Chen. He provided the opportunity for me to begin my postgraduate studies sooner than I'd thought was possible, and has supported me throughout my time in Liverpool. He has also offered me my first postdoctoral research post, for which I am very grateful.

Abstract

The work in this thesis is concerned with variational methods for two-phase segmentation problems. We are interested in both the obtaining of numerical solutions to the partial differential equations arising from the minimisation of a given functional, and forming variational models that tackle some practical problem in segmentation (e.g. incorporating prior knowledge, dealing with intensity inhomogeneity). With that in mind we will discuss each aspect of the work as follows.

A seminal two-phase variational segmentation problem in the literature is that of Active Contours Without Edges [33], introduced by Chan and Vese in 2001, based on the piecewise-constant formulation of Mumford and Shah [89]. The idea is to partition an image into two regions of homogeneous intensity. However, despite the extensive success of this work its reliance on the level set method [95] means that it is nonconvex. Later work on the convex reformulation of [33] by Chan, Esedoglu, and Nikolova [30] has led to a burgeoning of related methods, known as the convex relaxation approach [78, 137, 25, 102]. In Chapter 4, we introduce a method to find global minimisers of a general two-phase segmentation problem, which forms the basis for work in the rest of the thesis. We introduce an improved additive operator splitting (AOS) method based on the work of Weickert et al. [129] and Tai et al. [85]. AOS has been frequently used for segmentation problems [105, 104, 9], but not in the convex relaxation setting. The adjustment made accounts for how to impose the relaxed binary constraint, fundamental to this approach. Our method is analogous to work such as Bresson et al. [18] and we quantitatively compare our method against this by using a number of appropriate metrics.

Having dealt with globally convex segmentation (GCS) for the general case in Chapter 4, we then bear in mind two important considerations. Firstly, we discuss the matter of selective segmentation and how it relates to GCS. Many recent models have incorporated user input for two-phase formulations using piecewise-constant fitting terms [105, 104]. In Chapter 5 we discuss the conditions for models of this type to be reformulated in a similar way to [30]. We then propose a new model compatible with convex relaxation methods, and present results for challenging examples. Secondly, we consider the incorporation of priors for GCS in Chapter 8. Here, the intention is to select objects in an image of a similar shape to a given prior. We consider the most appropriate way to represent shape priors in a variational formulation, and the potential applications of our approach.

We also investigate the problem of segmentation where the observed data is challenging. We consider two cases in this thesis; in one there is significant intensity

inhomogeneity, and in the other the image has been corrupted by unknown blur. The first has been widely studied [82, 34, 100] and is closely related to the piecewise-smooth formulation of Mumford and Shah [89]. In Chapter 6 we discuss a Variant Mumford-Shah Model by D.Chen et al. [37] that uses the bias field framework [49, 2]. Our work focuses on improving results for methods of this type. The second has been less widely studied, but is more commonly considered when there is knowledge of the blur type [69, 107]. We discuss the advantages of simultaneously reconstructing and segmenting the image, rather than treating each problem separately and compare our method against comparable models [11].

The aim of this thesis is to develop new variational methods for two-phase image segmentation, with potential applications in mind. We also consider new schemes to compute numerical solutions for generalised segmentation problems. With both approaches we focus on convex relaxation methods, and consider the challenges of formulating segmentation problems in this manner. Where possible we compare our ideas against current approaches to determine quantifiable improvements, particularly with respect to accuracy and reliability.

Publications

- J. Spencer and K. Chen. A convex and selective variational model for image segmentation. *Communications in Mathematical Sciences*, 13(6):1453-1472, 2015.
- J. Spencer and K. Chen. Stabilised bias field: Segmentation with intensity inhomogeneity. *Journal of Algorithms & Computational Technology*, 2016. submitted.
- J. Spencer and K. Chen. Advanced methods in variational learning: Segmentation with intensity inhomogeneity. In *Mathematical Problems in Data Science*, Springer, pages 171-187. Springer, 2015.
- J. Spencer and K. Chen. Global and local segmentation of images by geometry preserving variational models and their algorithms. In *Forging Connections between Computational Mathematics and Computational Geometry*, pages 87-105. Springer, 2015.
- B. Williams, J. Spencer, K. Chen, Y. Zheng, and S. Harding. An effective variational model for simultaneous reconstruction and segmentation of blurred images. *Journal of Algorithms & Computational Technology*, 2016. submitted.

Presentations

- Image Segmentation with a Shape Prior. SIAM Conference on Imaging Science, Albuquerque, 2016.
- Shape Prior Segmentation with Intensity Inhomogeneity. New Directions in Numerical Computation, University of Oxford, 2015.
- Selective Image Segmentation with Intensity Inhomogeneity. 26th Biennial Numerical Analysis Conference, University of Strathclyde, 2015.
- Dynamic Distance Selective Segmentation. Challenges in Dynamic Imaging Data, Isaac Newton Institute, 2015.

List of Acronyms

BV	bounded variation
TV	total variation
PDE	partial differential equation
AOS	additive operator splitting
GCS	globally convex segmentation
SSD	sum of squared differences
ROF	Rudin, Osher, and Fatemi
CV	Chan-Vese
CDF	Chambolle's dual formulation
DSS	distance selective segmentation
CDSS	convex distance selective segmentation
VMS	variant Mumford-Shah
SBF	stabilised bias field
GCV	TV deblurring followed by CV segmentation
PCV	Poisson deblurring followed by CV segmentation
BSK	Bar, Sochen, and Kiryati
2SG	implicitly constrained TV deblurring followed by GCS
2SP	implicitly constrained Poisson deblurring followed by GCS
JRS	joint reconstruction and segmentation
RRS	relaxed joint reconstruction and segmentation
FSP	fitting shape prior
DSP	distance shape prior

List of Figures

2.1	Illustration of the interface representation with the level set method. i) shows a function ϕ (conventionally a distance function) and its intersection with $\phi = 0$. ii) shows the corresponding zero level set of ϕ , which implicitly defines the interface based on the values of ϕ . The level set function is almost arbitrary (excepting possible numerically difficult choices) in this context as long as Γ remains unchanged.	23
3.1	Illustration of the interface representation in the convex relaxation framework. i) shows a binary function u . ii) shows the corresponding contour, which implicitly defines the interface. In the convex relaxation framework the binary constraint is relaxed, and Γ is given by a thresholding procedure for a parameter $\gamma \in (0, 1)$	41
3.2	Edge detection function, $g(x)$, from eqn. (3.2) for an image, $z(x)$	43
3.3	Approximation to the Heaviside and Delta functions with H_ϵ and δ_ϵ	45
4.1	The penalty function $\nu(u)$ used in [17, 29, 30] to enforce the constraint $u \in BV(\Omega; [0, 1])$	56
4.2	The regularised penalty function $\nu_\epsilon(u)$ for ii) $\epsilon = 1$, iii) $\epsilon = 0.1$, and iv) $\epsilon = 0.01$. The original penalty function, $\nu(u)$, from [30] is shown in i). Visually, the most appropriate choice is for $\epsilon = 0.01$	57
4.3	The second derivative of $\mathcal{P}(u)$, $\mathcal{P}_\epsilon''(u)$, given by (4.11). i) is for $\epsilon = 1$ and ii) is for $\epsilon = 0.1$. Both are non-negative and therefore the corresponding $\mathcal{F}_\epsilon(u)$ is a convex functional.	59
4.4	The function $\nu'_\epsilon(u)$ for different choices of ϵ . The jumps at $u = 0$ and $u = 1$ aren't as sharp for larger ϵ , but the constraint $u \in [0, 1]$ is enforced less strictly in these cases.	61
4.5	Test Problems. Two examples are given for two-phase segmentation problems where the ground truth is known. Image 1 and 2 are on the left and right, respectively. Row 1 is the observed image, row 2 is the fitting function $f(x)$, and row 3 gives the zero contour, Γ_f , of $f(x)$ in red.	63

4.6	Test Set 1. ϵ Results, Image 1. The left column is the segmentation function, $u^*(x)$, the central column is the histogram of $u^*(x)$, and the right column is the regularised penalty function, $\nu_\epsilon(u)$. Row 1 is for $\epsilon = 10^0$, row 2 is for $\epsilon = 10^{-1}$, row 3 is for $\epsilon = 10^{-2}$. This demonstrates that a good choice for ϵ in the regularised penalty function, $\nu_\epsilon(u)$, is 10^{-2} . This is consistent throughout our tests, including for Image 2 which is not shown here.	65
4.7	Test Set 1. AOS0 Results, Image 1. The left column is the segmentation function, $u^*(x)$, the central column is the histogram of $u^*(x)$, and the right column is the residual progression. Row 1 is for $\tau = 10^{-3}$, row 2 is for $\tau = 10^{-2}$, row 3 is for $\tau = 10^{-1}$. This demonstrates that for AOS0 a small time step ($\tau = 10^{-3}$) is required for a result that is close to binary, and a smooth convergence for $u(x)$. There are similar results for Image 2, which are not shown here.	67
4.8	Test Set 1. AOS1 ς Results, Image 1. In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). Residuals are presented for AOS1 results for Image 1 with $\tau = 10^{-2}$, for four different choices of ς . They demonstrate that the convergence for $u(x)$ is dependent on the width of I_ς , and it is possible to use larger time steps with the improved scheme, AOS1.	68
4.9	Test Set 1. AOS2 ς Results, Image 1. In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). Residuals are presented for AOS2 results for Image 1 with $\tau = 1$, for three different choices of ς . On the left is the segmentation function, $u^*(x)$, and on the right is the residual progression. They demonstrate that the convergence for $u(x)$ is dependent on the width of I_ς , and it is possible to use arbitrary time steps with the improved scheme, AOS2, when $\varsigma = 0.5$	69
4.10	Test Set 2. Accuracy Results, Image 1. Row 1 is the computed contour Γ^* (given in red on $z(x)$), and the right is the segmentation function $u^*(x)$. On the left are AOS2 results, and the right are CDF results (both for $\lambda = 5$). The plot shows the TC value when λ is varied for AOS2 and CDF. This demonstrates that whilst their best results are similar, AOS2 is successful for a much larger range of the fitting parameter.	70
4.11	Test Set 2. Binary Measurement, Image 1. Row 1 is the segmentation function $u^*(x)$, and row 2 is the histogram for $u^*(x)$. On the left are AOS2 results, and the right are CDF results (both for $\lambda = 5$). The plot shows the mb value when λ is varied for AOS2 and CDF. This demonstrates that AOS2 is consistently closer to a binary result than CDF.	71

4.12	Test Set 2. Accuracy Results, Image 2. Row 1 is the computed contour Γ^* (given in red on $z(x)$), and row 2 is the segmentation function $u^*(x)$. On the left are AOS2 results, and the right are CDF results (both for $\lambda = 5$). The plot shows the TC value when λ is varied for AOS2 and CDF. This demonstrates that whilst their best results are similar, AOS2 is successful for a much larger range of the fitting parameter.	72
4.13	Test Set 2. Binary Measurement, Image 2. Row 1 is the segmentation function $u^*(x)$, and row 2 is the histogram for $u^*(x)$. On the left are AOS2 results, and the right are CDF results (both for $\lambda = 5$). The plot shows the mb value when λ is varied for AOS2 and CDF. This demonstrates that AOS2 is consistently closer to a binary result than CDF.	73
4.14	Test Set 3. Initialisations, Image 1. Rows 1-4 are for initialisations I_1 - I_4 respectively. On the left is the initial segmentation function $u_0(x)$, and on the right is the initial contour Γ_0 on $u_0(x)$ in red.	75
4.15	Test Set 3. Initialisations, Image 2. Rows 1-4 are for initialisations I_1 - I_4 respectively. On the left is the initial segmentation function $u_0(x)$, and on the right is the initial contour Γ_0 on $u_0(x)$ in red.	76
4.16	Test Set 3. Initialisation Results. AOS1 results from Table 4.5 with $\delta = 0.01$. The top row is different views of the segmentation function $u^*(x)$ for $\varsigma = 0.5$ and initialisation I_1 . Here $mb = 99$ and $cpu = 3.3$. The bottom row is similar for initialisation I_2 . Here $mb = 85$ and $cpu = 98.4$. This demonstrates that initialising the segmentation function as close to the final result as possible offers significant advantages, both in terms of time it takes to reach the stopping criterion and how close to binary the result is when that happens. This is an example of the observations that can be made from Tables 4.4 and 4.5.	78
5.1	Test Set 1. Results for Rada-Chen [105], for three test problems (given by rows 1-3). From left to right: initialisation (with user input set \mathcal{A}), final contour, object selected	92
5.2	Test Set 1. Results for DSS, for three test problems (given by rows 1-3). From left to right: initialisation (with user input set \mathcal{A}), final contour, object selected	93
5.3	Test Set 2. Results for CDSS, for three test problems (given by rows 1-3). From left to right: initialisation (with user input set \mathcal{A}), final contour, object selected.	94
5.4	Test Set 2. User input set 1 for CDSS. From left to right, top to bottom: initialisation, P_d function (with user input set \mathcal{A}_4), final contour, object selected.	95
5.5	Test Set 2. User input set 2 for CDSS. From left to right, top to bottom: initialisation, P_d function (with user input set \mathcal{A}_5), final contour, object selected.	96

5.6	Test Set 3. Results for $AOS_1, \tau = 10^{-2}$ for CDSS. Row 1 is for $\zeta = 0.01$, row 2 is for $\zeta = 0.1$. From left to right: final contour and residual for u (with number of iterations).	97
5.7	Test Set 3. Results for $AOS_2, \tau = 1$ for CDSS. Row 1 is for $\zeta = 0.1$, row 2 is for $\zeta = 0.5$. From left to right: final contour and residual for u (with number of iterations).	98
5.8	Test Set 4, Image 1. Results for CDSS. i) Observed Image, $z(x)$. ii) The converged segmentation function, $u^*(x)$. iii) The edge detection function, $g(x)$, for the weighted TV term. iv) An alternative view of the segmentation function, $u^*(x)$. v) The distance selection term, $P_d(x)$, based on the user input set \mathcal{A} . vi) The thresholded contour, Γ^* , demonstrating an accurate result for the object of interest.	99
5.9	Test Set 4, Image 2. Results for CDSS. i) Observed Image, $z(x)$. ii) The converged segmentation function, $u^*(x)$. iii) The edge detection function, $g(x)$, for the weighted TV term. iv) An alternative view of the segmentation function, $u^*(x)$. v) The distance selection term, $P_d(x)$, based on the user input set \mathcal{A} . vi) The thresholded contour, Γ^* , demonstrating an accurate result for the object of interest.	100
6.1	VMS Results. Convergence Behaviour. The first row shows the lack of convergence for the intensity constants, giving $c_1 = 9.1 \times 10^4$ and $c_2 = 6.4 \times 10^4$ after 1000 iterations. The second row shows the scale of the bias field, $B(x)$, on the left and the image, $z(x)$, and computed contour, Γ^* , on the right. The bottom row shows the progression of the energies E^B and E^F (6.7) (iterations on horizontal axis).	105
6.2	Images tested with SBF and compared to results of D. Chen et al. [37].	110
6.3	SBF Set 1 Results. SBF convergence behaviour (see Section 6.3.4.) Rows 1-4 are for Images 1-4 respectively. From left to right: $z(x)$ and Γ^* computed with SBF, bias field $\tilde{B}(x)$, and the progression of c_1 values (vertical axis) against iterations (horizontal axis.) Similar behaviour for c_2 values is also observed.	112
6.4	SBF Set 1 Results. Piecewise-smooth Mumford-Shah approximation of Images 1-4, given from top to bottom. On the left is z , and on the right is the SBF piecewise-smooth Mumford-Shah approximation, given by w^{SBF} (6.13).	113

6.5	SBF Set 2 Results. SBF compared to VMS (see Section 6.3.4.) i) Successful segmentation of the image, $z(x)$, given by contour Γ^* . ii) Computed stabilised bias field, $\tilde{B}(x)$. iii) Convergence of c_1 values (50 iterations.) Similar behaviour for c_2 values is also observed. iv) Piecewise-smooth approximation of $z(x)$ with SBF. v) Piecewise-smooth approximation of $z(x)$ with VMS. vi) Difference between SBF and VMS approximations, demonstrating significant differences around Γ^* . vii) The TC measure for VMS (dotted red) and SBF (blue), demonstrating the segmentation quality falls away for VMS with large values of λ	115
6.6	SBF Set 2 Results. SBF compared to VMS (see Section 6.3.4.) The TC measure for $\lambda \in [5, 100]$ shows that an optimal Γ^* can be computed for a larger range of λ with SBF than VMS.	116
6.7	Selective SBF Results (see Section 6.4.1.) i) Successful selective segmentation of the image, $z(x)$, given by contour Γ^* . ii) Distance selection function, $P_d(x)$, with user markers. iii) Convergence of c_1 values (200 iterations.) iv) Convergence of c_2 values (200 iterations.) v) Computed stabilised bias field, $\tilde{B}(x)$. vi) Computed indicator function, $u(x)$	118
7.1	Images used for test examples.	131
7.2	Test Sets 1 and 2. Illustration of the performance of CV for Image 1 corrupted by Gaussian blur: i) Initial contour. ii) Segmentation given by CV. iii)-iv) segmentation given by JRS. CV gives a rough segmentation while the spaces between the letters which are hidden by the blur are successfully segmented using JRS.	135
7.3	Test Set 2. Illustration of the performance of the JRS for (top-bottom) Image 1, 2, 4, and 6 corrupted by strong Gaussian blur. JRS is capable of segmenting edges in these challenging cases which cannot be segmented by CV.	137
7.4	Test Set 2. Illustration of the performance of the JRS for (top-bottom) Image 1,3,4, and 5 corrupted by Gaussian blur and noise. The edges hidden by blur are successfully segmented by JRS which cannot be segmented by CV.	138
7.5	Test Sets 2 and 3. Illustration of the performance of JRS for (top-bottom) Image 3,4,5, and 6 corrupted by Gaussian blur. The edges hidden by blur are successfully segmented by JRS which cannot be segmented by CV.	140
7.6	Test Sets 2 and 3. Illustration of the performance of the JRS for Image 2 corrupted by Gaussian blur: i) Received data. ii)-iii) Segmentation using JRS. iv) the difference between the segmentation using JRS and using CV. The segmentation is closer to the true edge using JRS while CV also captures the blurred edge.	141
7.7	Test Set 3. Images corrupted by Gaussian blur segmented using RRS.	142

8.1	Chan and Zhu [36] and Pock et al. [130] represent shapes as distance functions, $\psi(x)$. The shape is implicitly defined as the zero level set of $\psi(x)$ (given in red).	146
8.2	The fitting term, $f(z)$, for a given image, z , based on known intensity constants c_1 and c_2 . Here, $\Gamma_f = \{x : f(z) = 0\}$ is given in red in iii) and approximates the shape of the object in z .	147
8.3	The shape prior term, $S(x)$, based on the prior image z_p . Here, $\Gamma_S = \{x : S(x) > \omega\}$ is given in red (for small ω) in iv) and approximates the shape of the object in z_p . The template is formed in this way such that $S(x) \in [0, 1]$.	148
8.4	The choice of proximity function: $P_\beta = 1 - \frac{1}{\beta} \min\{\beta, d(\tilde{u})\}$, where $d(\tilde{u})$ is the normalised Euclidean distance from the object prior. This forms the reference for Stage 1 (Affine registration), $P_\beta(\tilde{u})F(z)$. i) The translated binary prior, \tilde{u} , given by (8.8). ii-iv) The function $P_\beta(\tilde{u})$ for $\beta = 1, 0.5$ and 0.1 respectively.	151
8.5	Test Set 1 (Occlusions). i) An image with an artificial occlusion. We refer to this example as Occlusion 1. ii) The same image with a different artificial occlusion. We refer to this example as Occlusion 2. iii) Shape prior term, $S(x) = -H(-f(z_p)f(z_p))$, based on our method of using the fitting term of a similar image to construct an approximate shape representation. iv) A comparison between \tilde{u} , a translation of u_p , and the boundary of the ground truth of z given by Γ_{GT} (red).	156
8.6	Test Set 1 (Occlusions). Stage 1: affine registration from Section 8.4.1. i) In the affine registration framework the shape prior, $S(x)$, forms the template. ii) The fitting term from the observed data, $P_\beta(\tilde{u})F(z)$, forms the reference. iii) The image, z , for Occlusion 1. iv) The result of Stage 1, where the parameters \mathbf{a}^* have been found, giving $S(\phi^*)$. This forms the basis for Stage 2.	157
8.7	Test Set 1 (Occlusions). Stage 2: segmentation from Section 8.4.2. i) Shape prior term, $S(x)$, based on our method of using the fitting term of a similar image to construct an approximate shape representation. ii) The fitting term constructed from the registered prior, $S(\phi^*)$, given by $h(x)$ (8.18). iii) The computed contour, Γ^* , from Stage 2 of our algorithm. iv) The computed function $u^*(x)$ from the minimisation in the convex relaxation framework (8.19).	158
8.8	Test Set 1 (Occlusions). Stages 1 and 2 for Occlusion 2. i) The image, z , for Occlusion 2. ii) Shape prior term, $S(x)$, based on our method of using the fitting term of a similar image to construct an approximate shape representation. iii) Stage 1, where the parameters \mathbf{a}^* have been found, giving $S(\phi^*)$. iv) The fitting term constructed from $S(\phi^*)$, given by $h(x)$ (8.18). v) The computed contour, Γ^* , from Stage 2. vi) The computed function $u^*(x)$ (8.19).	159

8.9	Test Set 2 (Parameter Dependence). i) The prior image, z_p , from which we know u_p . ii) The target image, z , which we want to segment based on the shape of u_p . iii) Shape prior term, $S(x) = -H(-f(z_p)f(z_p))$, based on our method of using the fitting term of a similar image to construct an approximate shape representation. iv) A comparison between \tilde{u} , a translation of u_p , and the boundary of the ground truth of z given by Γ_{GT} (red).	160
8.10	Test Set 2 (Parameter Dependence). i) The result of Stage 1 of our algorithm, where $S(\phi^*)$ is determined based on the minimisation of the affine registration formulation (8.13). ii) In Stage 2 we construct a fitting term based on the shape and intensity of the object, given by $h(x)$ (8.18). iii) The computed contour, Γ^* , from Stage 2 of our algorithm. iv) The computed function $u^*(x)$ from the minimisation in the convex relaxation framework (8.19).	161
8.11	Test Set 2 (Parameter Dependence). Results obtained using DSP formulation. i) The binary shape, \tilde{u} , which is the ground truth of z_p . ii) An alternative prior, $\psi(\tilde{u})$, based on the Euclidean distance from the boundary of the translated prior. This term is similar to shape representations in [36, 130]. Here, $\Gamma_\psi = \{x : \psi(x) = 0\}$ and is shown in red. The computed contour, Γ^* , using DSP. iv) The computed function $u^*(x)$ from the minimisation problem of DSP (8.23).	162
8.12	Test Set 2 (Parameter Dependence). DSP compared against initial TC of \tilde{u} . i) Shape prior, $\psi(x)$, used in (8.23). Here, $\Gamma_\psi = \{x : \psi(x) = 0\}$ and is shown in red. ii) - iv) $TC(\lambda)$ for different choices of θ in DSP, and the initial TC of \tilde{u} . Varying $\lambda \in [0, 300]$ gives some improvement over the initial TC . As θ increases, the range of λ that offers an improvement gets larger. However, the extent of this improvement is also lessened as θ increases for $\lambda \in [0, 300]$. This makes sense as the $\psi(x)$ term favours \tilde{u} . Balancing λ and θ with DSP can be challenging, and offers limited improvements over the given prior.	163
8.13	Test Set 2 (Parameter Dependence). Two-Stage Fitting Shape Prior Model (FSP) compared against alternative DSP and the initial TC of \tilde{u} . i) Shape prior, $S(x)$, used in (8.10). ii) - iv) $TC(\lambda)$ for different choices of α in FSP against DSP (with $\theta = 300$), and TC of \tilde{u} . For $\alpha < 1$, TC is consistently above the initial TC , and peaks higher than DSP does for any λ, θ pair. For $\alpha = 0.1$, we can see that we have a substantial gain over DSP, both in terms of the optimal choice and the dependence on the parameters selection.	164

8.14	Test Set 3 (Sequential Selection). Problem Definition: Given a prior image, $z_p(x)$, and a corresponding shape prior, $S(x)$, according to Section 8.4 given by i) and ii) respectively, we aim to successfully segment the same object in a different slice of a 3D data set. iii) gives the target image, $z(x)$, and iv) gives the fitting term of $z(x)$. We can achieve a result by applying our proposed two-stage model to the intermediate slices, which is defined in detail in Algorithm 8.	165
8.15	Test Set 3 (Sequential Selection). Stage 1 Results. i) The image $z(x)$ at Slice 107 of the set. ii) The fitting function $h(x)$ determined from Stage 1 of the algorithm for Slice 107. Similar for iii)-iv) Slice 112, v)-vi) Slice 118, and vii)-viii) Slice 123.	166
8.16	Test Set 3 (Sequential Selection). Stage 2 Results. i) The computed contour Γ^* for Slice 107. ii) The segmentation function, $u^*(x)$, determined from Stage 2 of the algorithm. Similar for iii)-iv) Slice 112, v)-vi) Slice 118, and vii)-viii) Slice 123.	167

List of Tables

- 4.1 Test Set 1. AOS1 ς Results for Images 1 (128x128) and 2 (256x256).
 In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). We present values as ς varies in terms of segmentation quality, TC , a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion $\delta = 0.01$, cpu (n.b. for some results the iterations were stopped at the maximum iteration number). Results demonstrate that smaller values of ς produce non-binary results and take longer to converge, despite the accuracy of the thresholding procedure. 66
- 4.2 Test Set 2. AOS1 and CDF Results for Image 1 (128x128). We present values as λ (the fitting function parameter) varies in terms of segmentation quality, TC , a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion δ , cpu . For CDF we test two stopping criteria, $\delta = 0.1, 0.01$, and for AOS1 we test $\delta = 0.01$. Results demonstrate that AOS1 ($\varsigma = 0.05$) converges faster than CDF, and produces better results in terms of TC and mb for a range of λ 68
- 4.3 Test Set 2. AOS1 and CDF Results for Image 2 (256x256). We present values as λ (the fitting function parameter) varies in terms of segmentation quality, TC , a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion δ , cpu . For CDF we test two stopping criteria, $\delta = 0.1, 0.01$, and for AOS1 we test $\delta = 0.01$. Results demonstrate that AOS1 ($\varsigma = 0.05$) converges faster than CDF, and produces better results in terms of TC and mb for a range of λ 74
- 4.4 Test Set 3. Initialisation Results (AOS1), Image 1 (128x128). In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). We present values as ς varies in terms of a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion $\delta = 0.01$, cpu (n.b. for some results the iterations were stopped at the maximum iteration number). Four initialisations are used (shown in Fig. 4.14). Results demonstrate that varying ς affects the convergence time, depending on the choice of initialisation. One notes that whilst $\varsigma = 0.5$ makes convergence likely it can be slower than smaller values. 74

4.5	Test Set 3. Initialisation Results (AOS1), Image 2 (256x256). In the improved AOS scheme ζ determines the width of the interval, I_ζ (4.16). We present values as ζ varies in terms of a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion $\delta = 0.01$, cpu (n.b. for some results the iterations were stopped at the maximum iteration number). Four initialisations are used (shown in Fig. 4.15). Results demonstrate that varying ζ affects the convergence time, depending on the choice of initialisation. One notes that the best choice of ζ is not consistent for different initialisations.	77
7.1	Test Set 1. Error values for Images 1-6 corrupted by Gaussian blur and segmented by CV. In many cases, the competition is close but 2SG obtains the same or improved error values over competing models in all cases.	133
7.2	Test Set 1. Error values for Images 1-6 corrupted by Gaussian blur and segmented by PCV and 2SP. The competition is close for most examples, but overall 2SP outperforms PCV.	134
7.3	Test Set 1. Error values given by $L2A$ for Images 1-4 corrupted by Gaussian blur and segmented by 2SG, JRS and RRS. For Image 1, 2SG outperforms the other models but in the remaining cases JRS and RRS obtain improved results.	134
7.4	Test Set 2. Error values and cpu times (in seconds) for images Images 1-8 corrupted by strong Gaussian blur. In all cases, JRS and RRS achieve improved results and competition is close between JRS and RRS. For most cases, the cpu time is lower for RRS with the exception of three examples which have slightly lower cpu time for CV with deteriorated results.	136
7.5	Test Set 2. Error values and cpu times (in seconds) for Images 1, 3-5 corrupted by Gaussian blur and noise. In all cases, JRS and RRS achieve improved results. cpu time is lower for RRS in two cases. In the remaining cases, it is lower for CV and closely followed by RRS which achieved significantly improved results.	136
7.6	Test Sets 2 and 3. Error values and cpu times (in seconds) for images Images 1-8 corrupted by small Gaussian blur. In all cases, JRS and RRS achieve improved results with JRS typically achieving better results. For many examples, the cpu time is lower for CV but it is closely followed by RRS which gives considerably better results.	139

Chapter 1

Introduction

1.1 Image Segmentation

The subject of this thesis is the development of effective variational models for two-phase image segmentation, in the convex relaxation framework in particular. In brief, segmentation is the partitioning of an image into multiple regions of shared characteristics. The focus of this work is on the reliability of the result, and its robustness to parameter variation and user input in general. We are also concerned with the time taken to obtain a segmentation result, as minimising this is often essential in many applications.

In imaging there are essentially two different approaches: the discrete setting and the continuous setting. In the spatially discrete setting image pixels are assumed to be entities that are distinct from each other, whilst in the continuous setting images are defined as functions on a continuous domain. In relation to image segmentation, the aim in the discrete setting is to find an optimal labeling of each node (representing a pixel). Often the set of possible labels is binary (i.e. foreground/background), and a conventional approach is that of graph cuts where a global minimiser can be computed [62, 76]. This is a combinatorial method that can compute fast solutions, especially in the two dimensional case, but can suffer from accuracy limitations and difficulties in extending it to more challenging problems. A seminal approach in this setting is that of Geman and Geman [55] in 1984, which is closely related to the later work of Blake and Zisserman [14]. The continuous counterpart of Geman and Geman is the work of Mumford and Shah [89] in 1989. Much of the work in this thesis is based, at least in part, on this formulation of the segmentation problem [89], where the aim is to find a piecewise-smooth approximation of the image. The piecewise-constant formulation of Chan and Vese [33] is also of particular interest to this work.

In this thesis we concentrate on the continuous approach. Given an observed image, made up of pixels, the problem setting is the continuous domain where the aim is to determine a solution to an equation corresponding to the minimisation of a functional. Analytic solutions are very rare in this context, and so a numerical solution where the problem is discretised is required. This might seem counter-intuitive, but continuous methods have proven very successful since the seminal work of Mumford and Shah. Other significant developments since then include edge based methods and active con-

tour models. Noteworthy examples include the Snakes approach of Kass, Witkin, and Terzopoulos [72] and the Geodesic Active Contours model of Caselles, Kimmel, and Sapiro [22]. Important to the success of these approaches was the development of level set based methods [95, 143], which have been widely used over the last twenty years. It was utilised by Chan and Vese in the influential Active Contours Without Edges [33], a region based model based on the two-phase piecewise-constant Mumford-Shah formulation. The common theme with this approach to segmentation is that the problems are nonconvex, meaning that obtaining a global minimiser is often not possible.

Recent work addressing the issue of nonconvexity is based on the idea of convex relaxation, which is essential to the work in this thesis and will be discussed throughout. The original work in relation to segmentation in the continuous setting, is that of Chan, Esedoglu, and Nikolova [30] in 2006. This method aims to find the global minimiser of the two-phase piecewise-constant Mumford-Shah formulation, in the case of known intensity constants. The theoretical basis of this work is based on the work of Strang [119]. Related work since has included Bresson et al. [18] as well as many others [78, 137, 120, 25]. In short, the convex relaxation method consists of representing the regions within an image with a binary function, and relaxing this constraint such that it can take intermediate values. The partition between the regions is then given by a thresholding procedure. These approaches are generally formed of a fitting term, based on the observed data, and a regularisation term, typically based on the total variation seminorm.

We also concentrate on two-phase methods. That is, we want to partition the image into some meaningful foreground/background representation. This idea simplifies the segmentation problem significantly. Firstly, for the number of regions to be fixed is an advantage. An unsupervised segmentation where this has to be optimised is a challenging problem which has attracted attention recently, such as the work of Zhang et al. [141]. Secondly, multiphase (i.e. greater than two regions) problems are difficult in many respects, widely addressed in the literature. One notes as an aside that the analogous problem in the discrete setting is the Potts Model [103]. Multilabel problems of this type cannot be minimised globally with current discrete methods. Conventional approaches involve approximations of the harder problem, such as reducing it to a sequence of binary labeling problems [15]. Under certain conditions exact solutions can be computed, based on the work of Ishikawa and Geiger [67, 68]. Returning to the continuous setting; there have been a number of recent noteworthy developments [10, 19, 78, 79, 25, 136]. These are important to the content of this thesis as they are often generalisations of the two-phase methods we consider.

When partitioning an image into a foreground and background, we refer to a 'meaningful representation'. We now discuss what is meant by this and conventional approaches for achieving this distinction. The definition of meaningful depends on the problem setting, and the possible application. In a medical imaging context for example, it could mean identifying the boundary of an organ or tumour, or selecting a vessel. From a security perspective, it might mean selecting certain objects such

as vehicles or people. More generally, it is possible to classify certain characteristics that can determine the basis for the segmentation into categories: such as intensity [33, 34, 100], texture [144], or shape [101, 36, 46]. In our work we tend to focus on intensity, although we do address the inclusion of shape priors in Chapter 8. Within intensity based methods, there are also many possible approaches depending on the observed image. Broadly speaking, an image can be treated as piecewise-constant or piecewise-smooth, depending on the levels of intensity inhomogeneity present. Each type is closely related to the work of Mumford and Shah [89]. The former has proven very popular [33, 30, 18, 105, 20] and is effective for certain types of image. The latter has also attracted much attention recently [37, 122, 100, 34], and is applicable to a wider class of images. However, it is also more challenging as a constant can often be approximated without a priori knowledge of the image simplifying the piecewise-constant case in practice. We address the first problem in Chapter 6, and problems associated with a particular approach for images with intensity inhomogeneity.

It is important to note that image segmentation techniques can often fail based on limitations in the observed data. Such limitations can make an accurate segmentation difficult to determine without improving the quality of the data or providing additional information about the target object. These difficulties can take the form of poor image quality (i.e. the observed image contains significant levels of noise or blur), where locating the edge reliably is problematic. A possible solution to this is a pre-processing step where the quality of the image is improved before conventional segmentation methods are applied. Numerous variational approaches exist designed to improve the image quality; known as image restoration techniques. Noteworthy examples include the seminal work of Rudin, Osher, and Fatemi [109] in 1992 for total variation denoising, and blind deconvolution methods [35]. The limitations in the data can also take the form of incomplete data, either in the form of significant artefacts or occlusions. Again, many variational methods exist to improve the image quality in this case such as image inpainting [32]. A common practice is to incorporate prior knowledge of the target object into the model such that limitations in the observed data can be overcome. This can either be in the form of constraints [75], user input [92] and interaction [101], or alternate regularisation [106], as well as many others. We address these issues in Chapter 5 in relation to object selection, and Chapter 7 in relation to joint image restoration and segmentation.

In the following section we will outline the main chapters of the thesis, and then outline our contribution explicitly. Chapters 2 and 3 concern mathematical preliminaries and a background for variational methods in image processing. The remaining chapters all consist of original work, some of which has been published or presented in a similar form. However, all chapters contain supplementary results and discussion beyond previous versions of the work. In addition, the notation and nomenclature has been standardised where possible to make the content of the thesis easily understandable to the reader.

All of the work in this thesis is co-authored with my primary supervisor, Ke Chen.

The main idea of Chapter 4 has been published in [112], and presented at a conference last year [116], although all the results presented here are original. In Chapter 5 we present work previously published in [112] and [114] and presented in part at [113, 115]. The work contained in Chapter 6 has recently been submitted for publication [118] and an earlier version of it was published last year [111]. It has also been presented at two conferences [116, 115]. In Chapter 7 we present work which has been submitted for publication [131] in which I was not the primary author. It was joint work with Bryan Williams, Yalin Zheng, and Simon Harding, but has been amended and improved in order to be incorporated into the thesis. In Chapter 8 we present previously unpublished work, much of which was presented at SIAM Imaging Science 2016 [117].

1.2 Thesis Outline

Subsequent chapters of the thesis are organised as follows.

Chapter 2

In Chapter 2 we introduce some relevant mathematical preliminaries that will be useful in relation to the content of the later chapters. Subsequent chapters will refer to this review, and to the wider literature where necessary. This includes definitions and examples from mathematical areas such as normed linear 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, interface representation, and solving equations iteratively. The level of detail is necessarily low for brevity's sake, but it provides an overview of the essential details related to the subject of this thesis.

Chapter 3

Here, we provide a brief review of variational methods for image processing. We begin with related methods that are particularly useful for the work in this thesis. We introduce image denoising, and specifically the total variation (TV) model of Rudin, Osher, and Fatemi (ROF) [109]. This is a seminal work in the field and is closely related to the segmentation problems discussed in this thesis, primarily through the TV term. We also introduce ideas from image deblurring and registration which will be required later in the thesis. We then turn to the central idea of this chapter, reviewing segmentation methods essential to this work. We introduce convex relaxation methods which are considered throughout the later chapters of the thesis. We then briefly discuss algorithms that are applicable to variational imaging methods, with an emphasis on Chambolle's dual formulation [23] which is of particular interest to our work.

Chapter 4

In this chapter we focus on two-phase globally convex segmentation (GCS), with a generalised fitting function. We discuss recent work on convex relaxation methods in

relation to the problem we consider. We introduce a new penalty function to impose the relaxed binary constraint, $u \in [0, 1]$. Our main contribution is a new additive operator splitting (AOS) scheme, based on applying the work of Tai et al. [85] and Weickert et al. [128] to GCS. The methods we propose are intended to improve the quality of the result in two senses; first, the reliability of the thresholding procedure defined by Chan, Esedoglu, and Nikolova [30], and second, the accuracy of the final result in relation to a ground truth. We also aim to obtain improved results in relation to the computation time. Chapter 4 contains quantitative comparisons to an established method by Chambolle [23], where we examine the performance of these two methods with varied parameters. This chapter forms the basis for the rest of the work as it relates to the framework we use throughout the thesis.

Chapter 5

Having established a new approach for finding global minimisers for GCS, in Chapter 5 we address how fitting functions are determined in practice. We first introduce the concept of selective segmentation, where the intention is to select an image from within the foreground of a general two-phase approach. Conventionally, selective segmentation models tend to be level set based and thus finding global minimisers is problematic. We discuss the necessary conditions for selective segmentation models to be reformulated in a similar way to [30], discussed in detail in Chapters 3 and 4. With these in mind, we propose a new model and demonstrate its convex reformulation. We present experimental results intended to demonstrate the robustness of our approach to user input, both in the sense that minimal information is required and it can vary significantly. This is crucial for the potential applications of selective segmentation. We present results for difficult examples from medical imaging.

Chapter 6

In Chapter 6 we consider segmentation of images with significant intensity inhomogeneity. This requires a fitting function in GCS that goes beyond the piecewise-constant assumption of Chan-Vese [33]. This area has been widely studied in recent years [82, 100, 34, 3], particularly with relation to the piecewise-smooth formulation of Mumford and Shah [89]. A recent approach, based on the bias field framework, has proven an effective approach to approximating minimisers of the piecewise-smooth formulation [89]. Recent work by D. Chen et al [37], known as a Variant Mumford-Shah Model, is an example of segmentation model using bias field correction. We demonstrate contradictions in the formulation that prevent the convergence of the intensity constants, and introduce an additional constraint to improve the results. We discuss observed improvements with our stabilising method, and extend the idea to selective segmentation by incorporating the work from Chapter 5.

Chapter 7

In this chapter we consider the case where forming a fitting function based on the observed image data, as in Chapters 5 and 6, is not possible due to the image being corrupted by blur. Here, the image must be reconstructed before conventional segmentation methods can be applied. Many recent methods combine the ideas of image segmentation and deconvolution in the case where information about the blur is known [11, 28, 69]. However, the case where the blur is unknown has not seen many advances in recent years. We propose a joint model to simultaneously reconstruct and segment images corrupted by unknown blur, which we call blind image segmentation. Here, we combine implicitly constrained blind deconvolution and GCS. We also propose a relaxed method for accelerated convergence. We present results for a range of examples and compare our proposed methods to alternative approaches such as Bar et al. [11] and analogous two-stage approaches.

Chapter 8

In Chapter 8 we discuss the incorporation of shape priors for variational segmentation models. Specifically, we consider the most effective methods for including shape information in two-phase GCS. We review recent work in relation to this idea, and propose a new method to represent shapes based on the correspondence between the fitting functions of the prior and the observed image. We propose a two-stage model, incorporating affine registration, to segment objects of a similar shape to the prior. The results presented demonstrate the effectiveness of the proposed method in comparison to an analogous method using conventional shape representation techniques. We also consider the extension of our idea to 3D segmentation, with a sequential application of our algorithm.

1.3 Contribution

We conclude this section by discussing how our work contributes to the understanding of variational methods for image segmentation, and try to explain how each chapter is connected. In Chapter 4, we introduce a new method to compute minimisers of two-phase GCS problems and demonstrate practical and theoretical advantages over comparable methods. Firstly, in terms of the thresholding procedure inherent to convex relaxation methods, we present results that match the theoretical basis for the work more closely than the original work of Chan, Esedoglu, and Nikolova [30] and Bresson et al. [18]. In particular, our results are consistently closer to a binary result than in [30, 17, 18]. Whilst not offering an advantage in practice, it is a noteworthy improvement in relation to the underlying ideas of GCS. However, we also present results that demonstrate a quantifiable advantage over comparable methods in terms of performance. Specifically, the computation time and the parameter dependence is significantly improved with our approach to GCS with an improved AOS scheme compared to Chambolle’s dual formulation [23, 18]. We also highlight the importance of initialisation and discuss the

optimal choice in the context of GCS, which is often unaddressed in the literature.

In the later chapters we focus on some applications of the GCS framework, dealing with separate but related problems. The first is how to deal with challenging observed data, such as significant intensity inhomogeneity or blur. We deal with each problem in Chapters 6 and 7 respectively, proposing improvements in a theoretical and practical sense. In our work involving images with intensity inhomogeneity we address a contradiction in the widely used bias field framework [37, 49, 2]. Our proposed method allows us to reliably compute a result that is consistent with the observed image, and ensure that all variables involved converge. We also observe a reduction in the parameter dependence with our modification, offering an important advantage over existing methods. In Chapter 7, where we address images corrupted by blur, we discuss the benefits of reconstructing and segmenting the image simultaneously in the context of GCS. We compare our results against comparable two-stage methods, as well as existing methods [11], and conclude that this approach is effective.

Another important consideration in relation to the subject of the thesis is the incorporation of prior knowledge into GCS problems, which we address in Chapters 5 and 8. We consider two different problems; incorporating user input and data priors. Our approach to each is different based on the challenges involved in each area. In relation to segmentation with user input, which is generally referred to as selective segmentation [105, 104, 8], we consider how to improve the reliability of the models. Specifically, we discuss the conditions required to compute the global minimisers of such models by relating the problem to GCS. Previous approaches rely on local minima which often makes the quality of results unpredictable. We also demonstrate that our method is not sensitive to user input, which is vital for this type of model. In Chapter 8, we consider prior knowledge of a different form. Our contribution here consists of formulating the shape term by comparing data fitting terms rather than distance or binary based priors. Our results demonstrate that the segmentation quality is improved, as well as being less dependent on parameter choice over alternatives. We also consider extending this idea to 3D problems by treating the problem as a sequence of images, presenting some results for organ selection.

The work in Chapter 4 is applicable to any general two-phase GCS problem, including the problems presented in subsequent chapters, and we incorporate the ideas presented here throughout the thesis. It is worth noting that the problems discussed in Chapters 5-8 are also closely related. This is highlighted in Chapter 6, where we combine the considerations of the previous chapter to propose a selective segmentation model in the presence of intensity inhomogeneity. However, it is possible to consider problems that include aspects of each chapter and this work attempts to make these connections clearer. The methods proposed are applicable in a wide range of examples, and often address the principle underlying the problem of interest. We also focus on the practical advantages of our methods over established approaches, demonstrating significant quantifiable improvements.

Chapter 2

Mathematical Preliminaries

In this chapter we provide a brief summary of relevant mathematical preliminaries. Further to the discussion in Chapter 1 we introduce some concepts from linear vector spaces and some background for functions of bounded variation. We then discuss the setting for many image processing tasks, where we consider inverse problems requiring regularisation and the derivation of the corresponding partial differential equations using the theory of calculus of variations. We discuss the discretisation of these equations, such that a numerical solution to the original problem can be found. With respect to segmentation we discuss how an interface, corresponding with the unknown edge Γ , can be represented in a manner consistent with the discrete form of partial differential equations. Finally, we provide an overview of conventional methods for iteratively solving equations, both in the linear and nonlinear case. Further details can be found in the literature referenced throughout, and will also be addressed in later chapters. This chapter is intended to provide a brief summary of important mathematical theory that is essential to the work in this thesis.

2.1 Linear Vector Spaces

We begin by introducing the concept of a vector space, a basic mathematical structure formed by a collection of elements

$$u = (u_1, \dots, u_n),$$

called vectors. We then provide definitions that allow us to introduce normed linear spaces. Further detail can be found in the literature, such as [6].

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

1. If $u, v \in V$, then $u + v \in V$
2. $u + v = v + u$
3. $(u + v) + w = u + (v + w)$

4. There exists an element $\mathbf{0} \in V$, such that $u + \mathbf{0} = u$ for all $u \in V$
5. There exists an element $-u \in V$, such that $u + (-u) = \mathbf{0}$
6. For a scalar c , $cu \in V$
7. $c(u + v) = cu + cv$
8. $(b + c)u = bu + cu$
9. $b(cu) = (bc)u$
10. There exists an element $1 \in V$, such that $1u = u$ for all $u \in V$

Example 2.1.2 *Examples of linear vector spaces include*

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

2.1.1 Normed Linear Spaces

Definition 2.1.3 (Norm). Let $N : V \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ be a real valued function. Then N is a norm on V if it satisfies the following properties for all $u, v \in V$:

1. $N(u) = 0 \Rightarrow u = 0$,
2. $N(\alpha u) = |\alpha|N(u) \forall \alpha \in \mathbb{R}$,
3. $N(u + v) \leq N(u) + N(v)$.

Remark 2.1.4 *By the positive homogeneity axiom, we have $N(u) = N(-u)$. Along with the triangle inequality axiom we have positivity of the norm, i.e. $N(u) \geq 0$. When the first axiom does not hold, N is a seminorm on V .*

A norm induces a metric on V by

$$d(u, v) := N(u - v),$$

which is homogeneous and invariant under translations:

$$d(\alpha u, \alpha v) = |\alpha|d(u, v), \quad d(u + v, v + w) = d(u, w).$$

The norm of a vector u on the set of real numbers \mathbb{R} is usually represented by $\|u\|$.

Example 2.1.5 *Some important examples of norms:*

- ***p*-norm:**
Consider $u \in \mathbb{R}^n$, then for any real number $p \geq 1$ the *p*-norm of u is defined as

$$\|u\|_p = \left(\sum_{i=1}^n |u_i|^p \right)^{1/p}.$$

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

$$\|u\|_\infty = \max(|u_1|, |u_2|, \dots, |u_n|).$$

• **L^p -norm:**

Consider a continuous function f defined on a domain Ω such that

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

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

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

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

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

Definition 2.1.6 (Inner Product). An inner product on a linear vector space V is a function $\langle \cdot, \cdot \rangle_V$, defined on $V \times V$, which satisfies the following conditions (with scalar λ):

1. $\langle u, u \rangle_V > 0, \quad \forall u \neq 0$
2. $\langle u, v \rangle_V = \overline{\langle u, v \rangle_V}, \quad \forall u, v \in V$
3. $\langle \lambda u, v \rangle_V = \lambda \langle u, v \rangle_V, \quad \forall u, v \in V \text{ and } \forall \lambda$
4. $\langle u + v, w \rangle_V = \langle u, w \rangle_V + \langle v, w \rangle_V, \quad \forall u, v, w \in V$

Definition 2.1.7 (Normed Linear Space). If a vector space, V , is equipped with a norm $\|\cdot\|$ defined on it, then V is called a normed linear space.

Remark 2.1.8 A relevant example is Euclidean n -space (or Cartesian space), where the space of all n -tuples of real numbers $x \in \mathbb{R}^n$ is equipped with the Euclidean metric. A linear vector space with an inner product defined on it, is a special type of normed space. When a space is equipped with a seminorm, then it is called a seminormed linear space.

Definition 2.1.9 (Cauchy Sequence). Let $\{u_i\}$ be a sequence in a normed linear space V . This is a Cauchy sequence if for every $\varepsilon > 0$, there exists an $N \in \mathbb{N}$ such that

$$\|u_i - u_j\| < \varepsilon, \quad \forall i, j \geq N.$$

Definition 2.1.10 (Banach Space). A normed linear space V is said to be a Banach space if it is complete. That is, if every Cauchy sequence $\{u_i\} \subset V$ converges to an element $u \in V$.

Example 2.1.11 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 functions 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 2.1.12 (Hilbert Space). A space V with an inner product $\langle u, v \rangle$ such that every Cauchy sequence converges to an element of V , is called a Hilbert Space.

Definition 2.1.13 (Lipschitz Condition). *If for all $u, v \in S \subset \mathbb{R}$ for some $M \in \mathbb{R}$ the real function $f : S \rightarrow \mathbb{R}$ satisfies the Lipschitz condition in S :*

$$|f(u) - f(v)| \leq M|u - v|,$$

then f is called a Lipschitz continuous function.

The above definitions and examples cover some basic ideas essential to the variational methods discussed in this thesis. We will refer to this in later chapters, and discuss its relevance to the subject.

2.1.2 Convex Sets and Functions

We now introduce some important definitions and examples relating to convexity. These ideas are essential for understanding later chapters, as this is an important concept in relation to optimisation.

Definition 2.1.14 (Convex Set). *A set S in a vector space V is said to be convex if, for all $u, v \in S$ and all $\theta \in [0, 1]$, the point*

$$(1 - \theta)u + \theta v$$

is in S . In other words, every point on the line segment connecting u and v is in S .

Definition 2.1.15 (Convex Function). *A function $f : S \rightarrow \mathbb{R}$ defined on a convex set S of some vector space is called convex if*

$$f(\theta u + (1 - \theta)v) \leq \theta f(u) + (1 - \theta)f(v) \quad (2.1)$$

for all $u, v \in S$ and $\theta \in (0, 1)$. If the inequality is always strict for $u \neq v$, f is called strictly convex.

Theorem 2.1.16 *Let $I = (a, b)$ be an interval on \mathbb{R} . Then*

1. *A function f which is differentiable everywhere on I is convex on I if and only if its derivative is monotonically non-decreasing on I .*
2. *A function f which is twice differentiable everywhere on I is convex on I if and only if its second derivative is non-negative on I .*

Example 2.1.17 *The square of the L^2 -norm of a function $u : \Omega \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$ given by*

$$\|u\|_2^2 = \int_{\Omega} |u|^2 dx$$

is convex. By introducing a function ϕ and parameter ϵ , we can calculate the second derivative of a function $F(u)$ by making the substitution $v = u + \epsilon\phi$ and finding the second derivative with respect to ϵ :

$$\frac{d^2 F(v)}{d\epsilon^2} = \frac{d}{d\epsilon} \left(\frac{dF(v)}{dv} \frac{dv}{d\epsilon} \right) = \frac{d}{d\epsilon} \left(\frac{dF(v)}{dv} \phi \right) = \frac{d^2 F(v)}{dv^2} \frac{dv}{d\epsilon} \phi = \frac{d^2 F(v)}{dv^2} \phi^2.$$

Extending this to the L^2 -norm defined above:

$$\frac{d^2}{d\epsilon^2} \|u + \epsilon\phi\|_2^2 = \int_{\Omega} \frac{d^2}{d\epsilon^2} (u + \epsilon\phi)^2 dx = \int_{\Omega} \frac{d}{d\epsilon} 2\phi(u + \epsilon\phi) dx = \int_{\Omega} \phi^2 dx$$

we have demonstrated that the second derivative is non-negative. Then by the theorem, the square of the L^2 -norm of u is convex

Remark 2.1.18 Several operations preserve convexity, such as:

- *Weighted sums:* Let f and g be convex functions on \mathbb{R} . Then, the linear combination $h = \alpha f + \beta g$ is also convex for $\alpha, \beta \geq 0$.
- *Affine substitutions:* Let f be a convex function on \mathbb{R}^n and $\mathcal{A} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ be an affine mapping given by $\mathcal{A}(x) = Ax + b$. Then $f(\mathcal{A}(x))$ is also convex.

The following definition is important to understanding many imaging models, especially in the context of functions of bounded variation which we will come to next. We now define the subgradient of a function, with further detail found in the literature [24, 51].

Definition 2.1.19 (Subgradient). A function f is convex and defined on a finite dimensional space U . For $u \in U$,

$$\partial f(u) = \{p \in U : f(v) \geq f(u) + \langle p, v - u \rangle \forall v \in \text{dom} f\}$$

We note that $\text{dom } \partial f = \{u : \partial f(u) \neq \emptyset\} \subset \text{dom } f$, and if f is differentiable at u , then $\partial f(u) = \{\nabla f(u)\}$. It is also the case that $u \in \arg \min_U f$ if and only if $0 \in \partial f(u)$. This is evident, as it is equivalent to $f(v) \geq f(u) + \langle 0, v - u \rangle \forall v$. This idea is important to consider for the work in this thesis, as we are interested in convex functionals on a space of functions that is not necessarily differentiable.

2.2 Functions of Bounded Variation

In this section we introduce the idea of functions of bounded variation (BV). Functions of this type are important to many variational methods in imaging, due to total variation (TV) based regularisation which is common in many seminal works [109, 33, 30]. Further details can be found in the literature [57, 52, 53, 25, 1]. We begin by introducing some important definitions.

Definition 2.2.1 (Compact support). If $U \subset \mathbb{R}^n$ is nonempty, we denote \overline{U} as the closure of U in \mathbb{R}^n . We write $U \Subset \Omega$ if $\overline{U} \subset \Omega$ and \overline{U} is a closed and bounded (i.e. compact) subset of \mathbb{R}^n . If f is a function defined on U , the support of f is defined as the set

$$\text{supp}(f) = \overline{\{x \in U : f(x) \neq 0\}}.$$

The function f has compact support in Ω if $\text{supp}(f) \Subset \Omega$. As an example of the notation we use, the space of continuously differentiable functions, $C^1(\Omega)$, with compact support is denoted $C_0^1(\Omega)$.

Definition 2.2.2 (Total Variation (TV) seminorm). Let Ω be a bounded open subset of \mathbb{R}^n and $u \in L^1(\Omega)$. Define the total variation,

$$TV(u) = \int_{\Omega} |Du| = \sup_V \left\{ \int_{\Omega} u \operatorname{div} \varphi \, dx \right\}, \quad (2.2)$$

where V is the set of the test functions

$$V = \{ \varphi = (\varphi_1, \varphi_2, \dots, \varphi_n) \in C_0^1(\Omega; \mathbb{R}^n) : |\varphi(x)|_{L^\infty(\Omega)} \leq 1, \forall x \in \Omega \},$$

and

$$\operatorname{div} \varphi = \sum_{i=1}^n \frac{\partial \varphi_i}{\partial x_i}.$$

A noteworthy example is when $u \in C^1(\Omega)$, then

$$\int_{\Omega} u \operatorname{div} \varphi \, dx = - \int_{\Omega} \sum_{i=1}^n \frac{\partial u}{\partial x_i} \varphi_i \, dx$$

for every $\varphi \in C_0^1(\Omega; \mathbb{R}^n)$ using integration by parts. Then,

$$\int_{\Omega} |Du| = \int_{\Omega} |\nabla u| \, dx. \quad (2.3)$$

Remark 2.2.3 Using this definition, we can highlight an important property. Recalling the total variation of $u \in C^1(\Omega)$ as

$$TV(u) = \int_{\Omega} |\nabla u|, \quad (2.4)$$

we can show this is a convex function as follows. With $u_1 \neq u_2$:

$$\begin{aligned} \int_{\Omega} |\nabla(\alpha u_1 + (1 - \alpha)u_2)| &= \int_{\Omega} |\alpha \nabla u_1 + (1 - \alpha) \nabla u_2| \\ &\leq \alpha \int_{\Omega} |\nabla u_1| + (1 - \alpha) \int_{\Omega} |\nabla u_2| \end{aligned}$$

Therefore,

$$TV(\alpha u_1 + (1 - \alpha)u_2) \leq \alpha TV(u_1) + (1 - \alpha)TV(u_2)$$

which meets the definition from the previous section. This property of the TV seminorm is particularly important, and we will return to it later.

Definition 2.2.4 (Bounded Variation (BV)). If for a function $u \in L^1(\Omega)$, $TV(u) < \infty$, then the function u is said to be of bounded variation. The notation $BV(\Omega)$ denotes all functions in $L^1(\Omega)$ that are of bounded variation.

Remark 2.2.5 Under the norm

$$\|u\|_{BV} = \|u\|_{L^1} + \int_{\Omega} |Du|,$$

$BV(\Omega)$ is a Banach space.

2.2.1 Co-area Formula

A fundamental property of BV functions is the coarea formula of Federer and Fleming [52, 53]. It states that for a real-valued Lipschitz function $u(x)$ in an open set Ω in \mathbb{R}^n , the total variation of $u(x)$ can be computed by summing the perimeters of all level sets of $u(x)$. Formally, for $u(x) \in BV(\Omega)$ defined in Ω we define the level domain in \mathbb{R}^n , as

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

Then, for any continuous and integrable function $g(x) \in C^1$

$$\int_{\mathbb{R}^n} g(x) |\nabla u(x)| dx = \int_{-\infty}^{+\infty} \left(\int_{E_\gamma} g(x) ds \right) d\gamma.$$

For the particular case when $g(x) = 1$ and the region of integration is a subset $\Omega \subset \mathbb{R}^n$ we have

$$\int_{\Omega} |\nabla u| dx = \int_{-\infty}^{+\infty} \left(\int_{E_\gamma} ds \right) d\gamma = \int_{-\infty}^{\infty} d\gamma \int_{\Omega} |D\chi_{E_\gamma}| dx$$

Definition 2.2.6 (Perimeter). *The perimeter of $E_\gamma \in \Omega$ is defined as*

$$Per(E_\gamma) = \int_{\Omega} |D\chi_{E_\gamma}| = \sup_V \left\{ \int_{E_\gamma} \operatorname{div} \varphi \, dx \right\}, \quad (2.6)$$

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

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

Definition 2.2.7 (Coarea formula). *Assume that u is Lipschitz continuous and that for almost every $\gamma \in \mathbb{R}$. Then*

$$TV(u) = \int_{\Omega} |Du| = \int_{-\infty}^{\infty} Per(E_\lambda) d\lambda. \quad (2.7)$$

The proof can be found in [57]. An interesting characterisation of functions of bounded variation is as follows.

Example 2.2.8 *The TV of a characteristic function, χ_E , is given as follows*

$$\int_{\Omega} \chi_E \operatorname{div} \varphi \, dx = \int_E \operatorname{div} \varphi \, dx = \int_{\partial E} \vec{n} \cdot \varphi \, ds,$$

where \vec{n} is the outward unit normal to ∂E . The expression is maximised for any vector field with $\varphi|_{\partial E} = \vec{n}$, and hence

$$TV(\chi_E) = \int_{\partial E} ds = \mathcal{H}_{n-1} \quad (2.8)$$

where \mathcal{H}_{n-1} is the $(n-1)$ -dimensional Hausdorff measure, i.e. length when $n=2$.

This demonstrates that for $u \in BV(\Omega)$, the total variation of this function is the sum of the length of its level sets. In the context of imaging, this means that the discontinuities of the function and therefore the edges of the image are accounted for.

2.2.2 Derivative of a BV Function

We now briefly deal with an important theoretical aspect of this section; the derivative of functions of bounded variation. Further details can be found in Chambolle et al. [24], and the related literature. First, let's consider a simple case. For a function $u \in W^{1,1}(\Omega)$, then $Du = \nabla u(x)dx$, where ∇u is a vector-valued function in $L^1(\Omega; \mathbb{R}^n)$ called the "weak gradient". Then, Du is said to be absolutely continuous with respect to Lebesgue's measure. However, for a general function $u \in BV(\Omega)$, we have

$$Du = \nabla u(x)dx + D^s u,$$

where $D^s u$ is the singular part of Du , which vanishes if and only if $u \in W^{1,1}(\Omega)$. This can be further decomposed into the "jump set" and the "Cantor part", with further details given in [24, 57, 51]. In the rest of the thesis, despite the fact that we consider BV functions and cannot guarantee that $D^s u$ vanishes, we approximate Du with the weak gradient. Furthermore, in practice we deal with a discretised setting where we compute the gradient numerically in a similar way to related work [30, 109, 18].

2.3 Inverse Problems

In forward modeling problems, the aim is to compute solutions (approximate or exact) based on some known properties. An inverse problem begins with data that is the result of a process, and the task is to find the unknown input. In other words, it is the inverse of a forward problem. Inverse problems arise in many important fields, and improving our understanding of them is essential in numerous practical applications. Of particular interest to this work, we note important inverse problems in image processing, such as denoising [109], deblurring [35], inpainting [32], and registration [66]. A common consideration with inverse problems is that they are often ill-posed, which we will now discuss.

2.3.1 Well and Ill-Posed Problems

The definition of well-posedness, based on Hadamard [63], is given as follows.

Definition 2.3.1 (Well-posed Problem). *A problem is well-posed if the following conditions hold:*

- *A solution exists,*
- *The solution is unique,*
- *The solution depends continuously on the data.*

Typically, inverse problems are not well-posed and the stability condition above is most often violated. However, with many problems in image processing the uniqueness condition is not met. Problems that are not well-posed in the sense of Hadamard are considered ill-posed. An example of same is given as follows.

Example 2.3.2 *Image restoration problems such as denoising are well known, and this example highlights what is meant by an ill-posed problem in practice. Consider an image u defined in $\Omega \subset \mathbb{R}^2$, and an observed noisy image z that contains some additive Gaussian noise, η , such that the relation $z = u + \eta$ is satisfied. The inverse problem of finding u given z can be approached using variational methods, where an estimate of the standard deviation of the noise is assumed (σ^2). The minimisation problem is then:*

$$\min_u \left\{ \int_{\Omega} |u - z|^2 dx = \sigma^2 \right\}.$$

Here, there are many possible solutions, i.e. the solution is not unique. Therefore, this inverse problem is ill-posed.

2.3.2 Regularisation

In 1963 Tikhonov [121] introduced a seminal approach to transform an ill-posed inverse problem into a well-posed one, known as regularisation. This consists of introducing a new constraint that imposes certain restrictions on the solution. Typically this takes the form of penalising the complexity of the function, such as requiring smoothness. We will illustrate this approach in relation to Example 2.3.2, which is regularised in the following way:

$$\min_u \left\{ \int_{\Omega} |u - z|^2 dx + \alpha \int_{\Omega} |\nabla u|^2 dx \right\}. \quad (2.9)$$

The first term is the fitting term, that stipulates that the solution must closely resemble the observed data. The second is the regularisation term that requires the function u to have low gradient values, and α is a parameter that determines the level of noise reduction. The balance between the fitting and regularisation terms is empirically determined based on the properties of the desired solution. The solution to this minimisation problem is unique, and hence now it is well-posed.

In general, Tikhonov regularisation is defined as follows. Consider a given $A : D(A) \subseteq X \rightarrow Y$ operator between the Hilbert spaces X and Y such that $Au = b$. When a solution for u does not satisfy the conditions of well-posedness, Tikhonov [121] proposed minimising the following functional:

$$\min_u \left\{ \int_{\Omega} \|Au - b\|_2^2 dx + \alpha \int_{\Omega} \|u\|_2^2 dx \right\}. \quad (2.10)$$

This technique can be generalised by selecting alternate penalty functionals in the form of norms or seminorms of the function.

2.4 Calculus of Variations

In this section we introduce some essential concepts of the calculus of variations. Further detail can be found in the literature [54, 56, 7]. The idea is to find the optimal curve or surface based on an optimality criterion given in the form of a functional, which is a function of another function that assigns a real number to each function

in some class. We cover how this functional is minimised, by introducing the Gâteaux derivative and Gauss's Theorem, and provide an example of some of particular interest to this work.

2.4.1 Variation of a Functional

The first variation of a functional deals with the problem of finding a function for which the value of a certain integral is at its largest or smallest. Classical solutions to minimisation problems in the calculus of variations are given by boundary value problems involving certain types of differential equations. These are referred to as Euler-Lagrange equations. Consider the general functional $\mathcal{J}(u) : \Omega \rightarrow \mathbb{R}$

$$\mathcal{J}(u) = \int_{\Omega} F(x, u(x), \nabla u(x)) dx$$

where Ω denotes some normed linear space that is a solution space of the unknown function u , $\nabla u(x)$ denotes the gradient of u , and dx is the n -differential element defined as $dx = dx_1 \cdots dx_n$. We are concerned with the problem of minimising the functional $\mathcal{J}(u)$ with respect to u :

$$\min_u \mathcal{J}(u). \tag{2.11}$$

The most important necessary condition to be satisfied by any minimiser of a variational integral is the vanishing of its first variation $\delta\mathcal{J}(u)$:

$$\delta\mathcal{J}(u) = \left. \frac{d}{d\varepsilon} \mathcal{J}(u + \varepsilon\varphi) \right|_{\varepsilon=0} = 0 \tag{2.12}$$

where $\varphi \in \Omega$ is a test function and ε is a real parameter (restricted to some interval around 0). That is, if u is a minimiser of $\mathcal{J}(u)$ with respect to $\delta u = \varphi$, then (2.12) must be satisfied for all φ with compact support in Ω . Then we call $\delta\mathcal{J}(u_0)$ the first variation of \mathcal{J} at u_0 in the direction of φ , for some $u_0 \in \Omega$.

2.4.2 Gâteaux Derivative of a Functional

Definition 2.4.1 (Gâteaux Derivative). *Let \mathcal{J} be a function on an open subset U of a Banach space V , taking values in a second Banach space Y . Then we say $\mathcal{J} : U \rightarrow Y$ is Gâteaux differentiable at $u \in U$ in the direction of $\varphi \in V$, if the first directional derivative $\mathcal{J}'(u; \varphi)$ exists for each test function $\varphi \in V$. That is,*

$$\delta\mathcal{J}(u) = \lim_{\varepsilon \rightarrow 0} \frac{\mathcal{J}(u + \varepsilon\varphi) - \mathcal{J}(u)}{\varepsilon}.$$

Remark 2.4.2 *In other words the Gâteaux derivative of $\mathcal{J}(u)$ is just the derivative of $\mathcal{J}(u + \varepsilon\varphi)$ with respect to ε , evaluated at $\varepsilon = 0$.*

We now introduce some important definitions corresponding to the minimisation of a functional. The distinctions described below are fundamental to many aspects of this thesis, particularly in relation to Chapter 4.

Definition 2.4.3 (Stationary Point). Let $\mathcal{J} : U \rightarrow \mathbb{R}$ be a function with solution space $U \subset V$. For some $\tilde{u} \in U$, suppose \mathcal{J} is Gâteaux-differentiable for all test functions $\varphi \in V$. Then $\tilde{u} \in U$ is said to be a stationary point of \mathcal{J} if $\delta\mathcal{J}(\tilde{u}) = 0$ for all $\varphi \in V$.

Definition 2.4.4 (Local Minimiser). A real-valued functional $\mathcal{J} : U \rightarrow \mathbb{R}$, defined in the normed space V , is said to have a local minimiser at the point \tilde{u} , if there exists some $\epsilon > 0$ such that

$$\mathcal{J}(\tilde{u}) \leq \mathcal{J}(u), \quad \forall u \in B_\epsilon(\tilde{u}) \cap U,$$

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

Definition 2.4.5 (Global Minimiser). A real-valued functional $\mathcal{J} : U \rightarrow \mathbb{R}$ is said to have a global minimiser at the point \tilde{u} , if $\mathcal{J}(\tilde{u}) \leq \mathcal{J}(u)$, $\forall u \in U$.

Remark 2.4.6 The local and global maximisers of a functional $\mathcal{J}(u)$ can be defined in a similar way, by adjusting the inequalities.

The equation $\delta\mathcal{J}(u) = 0$ is called the Euler-Lagrange equation of the original minimisation problem (2.11). If $\mathcal{J}(u)$ is a convex functional, and U is a convex set, then every local minimiser of $\mathcal{J}(u)$ is also a global minimiser. This is a useful property, that we will return to later.

2.4.3 The Divergence Theorem

The Divergence Theorem, also known as Gauss's theorem, is essential to obtaining the Euler-Lagrange equation when minimising a functional. This idea will be referred to throughout the thesis, and is given as follows.

Theorem 2.4.7 (Gauss's Theorem). Let F be a continuously differentiable vector field in a domain $V \subset \mathbb{R}^n$. Let $\Omega \subset V$ be a closed, bounded region whose boundary, $\partial\Omega$, is smooth. The volume integral of the divergence of F over Ω and the surface integral of F over the boundary $\partial\Omega$ are then related by

$$\int_{\Omega} (\nabla \cdot F) dx = \int_{\partial\Omega} F \cdot \vec{n} ds.$$

where $\nabla \cdot F = \frac{\partial F}{\partial x_1} + \dots + \frac{\partial F}{\partial x_n}$, $dx = \{dx_1, \dots, dx_n\}$, ds indicates integration with respect to surface area on $\partial\Omega$, and \vec{n} is the unit outward normal for each point $x \in \partial\Omega$.

An important consequence of the Divergence Theorem can be seen by applying integration by parts to the product of a scalar function g and a vector field F . This gives us the following relation:

$$\int_{\Omega} (F \cdot \nabla g + g \nabla \cdot F) dx = \int_{\partial\Omega} g F \cdot \vec{n} ds. \quad (2.13)$$

We conclude this section by providing an illustrative example of the Divergence Theorem, with a particular application to variational segmentation. The relation above (2.13) will be referred to when determining the boundary conditions.

Example 2.4.8 Consider the problem of finding the first variation of the functional

$$\mathcal{J}(u) = \int_{\Omega} |\nabla u| \, dx,$$

defined on a domain $\Omega \subset \mathbb{R}^2$. Recall that $\varepsilon\varphi$ consists of the parameter $\varepsilon \rightarrow 0$ and the continuously differentiable test function φ in Ω . Then we compute,

$$\begin{aligned} \left. \frac{d}{d\varepsilon} \mathcal{J}(u + \varepsilon\varphi) \right|_{\varepsilon=0} &= \left. \frac{d}{d\varepsilon} \int_{\Omega} |\nabla(u + \varepsilon\varphi)| \, dx \right|_{\varepsilon=0} \\ &= \left. \int_{\Omega} \frac{\nabla(u + \varepsilon\varphi)}{|\nabla(u + \varepsilon\varphi)|} \cdot \nabla\varphi \, dx \right|_{\varepsilon=0} \\ &= \int_{\Omega} \frac{\nabla u}{|\nabla u|} \cdot \nabla\varphi \, dx. \end{aligned}$$

Using the relation introduced above (2.13), based on Gauss's Theorem and integration by parts, we get

$$\int_{\Omega} \frac{\nabla u}{|\nabla u|} \cdot \nabla\varphi \, dx = \int_{\partial\Omega} \varphi \frac{\nabla u}{|\nabla u|} \cdot \vec{n} \, ds - \int_{\Omega} \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \varphi \, dx.$$

We require

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

for all test functions φ . This allows us to derive the following partial differential equation, known as the Euler-Lagrange equation:

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

with Neumann boundary conditions, $\nabla u \cdot \vec{n} = 0$ on $\partial\Omega$.

2.5 Discretisation of Partial Differential Equations

Solving a partial differential equation (PDE) analytically is often not possible. Typically the Euler-Lagrange equations arising from variational imaging models fall into this category. As discussed in Chapter 1, we therefore attempt to obtain numerical solutions by solving a discrete version of the continuous PDE. A number of approaches exist to address this issue, such as the finite element method. However, in image processing problems the domain $\Omega \subset \mathbb{R}^2$ tends to be rectangular with uniformly distributed points. As a result, it is natural to use the finite difference method to discretise the domain. In this section, we will discuss some relevant details associated with problems of interest in this thesis.

We consider the domain $\Omega = (a, b) \times (c, d) \subset \mathbb{R}^2$, 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 directions respectively. In a cell-centered discretisation (as opposed to vertex-centered) there are $n_x \times n_y$ grid points and the point (i, j) is located at

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

We call the interior of the discrete grid Ω_h and the boundary $\partial\Omega_h$. Generally, we assume $h_x = h_y = h$. Now, operators from the PDE in the continuous domain can be approximated locally on Ω_h using the Taylor expansions

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

We now detail three possible approximations of the derivative $\partial u / \partial x$ at the grid point (i, j) . There is the forward difference operator

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

and the backward difference operator

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

where $u_{i,j} = u(x_i, y_j)$ is the value of $u(x, y)$ at the point (i, j) . These are both first order methods. A second order approximation based on central differences can be given as

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

We can also approximate higher order derivatives in a similar way. For example, a second order approximation of $\partial^2 u / \partial x^2$ at (i, j) is given by

$$\Delta^x(u_{i,j}) = \nabla_x^-(\nabla_x^+(u_{i,j})) = \frac{u_{i+1,j} - u_{i,j} + u_{i-1,j}}{h^2}.$$

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

With this in mind, it is often possible to write a PDE in the continuous domain in matrix form:

$$A_h(u_h) = f_h.$$

It is also important to consider how boundary conditions can be defined in the discrete domain. For cell-centered grids there are no points on the boundary so the equation at points near the boundary will involve ghost points. For example, to impose Neumann boundary conditions on one side of the domain we can write

$$\frac{u_{n+1,j} - u_{n,j}}{h} = f_{n+1/2,j}.$$

Specific examples of discretisations of nonlinear PDEs related to imaging problems are given in later chapters when relevant.

2.6 Interface Representation

In variational segmentation the aim is to find an interface, Γ , that partitions the domain, Ω , into distinct regions. With this in mind we briefly introduce some background of how this interface is represented in the literature.

2.6.1 Curves in Euclidean Spaces

With respect to the segmentation of images, we are concerned with a domain $\Omega \subseteq \mathbb{R}^2$. These ideas generalise to higher dimensions, but will not be addressed here. As such we limit this discussion to closed curves, with clearly defined interior and exterior regions. In simple cases, such as the boundary of the unit disk, the corresponding curve can be given analytically. However, typically it is necessary to parametrise the curve with a function $\phi = (x_1(t), x_2(t))$, for $t \in [a, b]$. To describe a closed curve implies $\phi(a) = \phi(b)$. The corresponding parametric equation for the boundary of the unit disk is then

$$\phi(t) = (\cos(t), \sin(t)), \quad \text{for } 0 \leq t < 2\pi.$$

When the interface cannot be given analytically, or when it evolves from some simple initialisation (which we will address in the following chapters) a parametric representation is not practical. We now provide some important definitions of geometric characteristics relating to the interface, and then introduce the Heaviside and Dirac delta functions. This will be useful in subsequent chapters.

Definition 2.6.1 For a scalar function $\phi(x_1, \dots, x_n)$ the gradient is denoted $\nabla\phi$ and is defined as

$$\nabla\phi = \left(\frac{\partial\phi}{\partial x_1}, \dots, \frac{\partial\phi}{\partial x_n} \right).$$

The gradient of ϕ points in the direction of increasing ϕ , perpendicular to its isocontours. The unit outward normal vector \vec{n} points in the same direction as $\nabla\phi$ for points on the interface, and is defined

$$\vec{n} = \frac{\nabla\phi}{|\nabla\phi|}.$$

Definition 2.6.2 The curvature of the interface is defined as the divergence of the unit normal \vec{n} , and is denoted κ :

$$\kappa = \text{div } \vec{n} = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} = \frac{\partial}{\partial x_1} \left(\frac{\nabla\phi}{|\nabla\phi|} \right) + \dots + \frac{\partial}{\partial x_n} \left(\frac{\nabla\phi}{|\nabla\phi|} \right).$$

2.6.2 Heaviside and Dirac delta function

The Heaviside function, often called the unit step function, is useful when dealing with piecewise continuous functions, i.e. functions containing sharp jumps. It is relevant here as it allows us to represent the interface in an alternative way.

Definition 2.6.3 *The Heaviside function, is a discontinuous function whose value is zero for negative arguments and one for positive arguments. For a given function $\phi(x)$, $x \in \mathbb{R}^n$:*

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

In one dimensional space, the derivative of the Heaviside is called the delta function, $\delta(\phi) = H'(\phi)$, and is zero everywhere except at $\phi = 0$. The characteristic function of the interior and exterior regions, denoted by χ_1 and χ_2 respectively, can be expressed in terms of the Heaviside as follows

$$\chi_1 = H(\phi) \quad \text{and} \quad \chi_2 = 1 - H(\phi).$$

Definition 2.6.4 *For a given function $\phi(x)$, $x \in \mathbb{R}^n$, the directional derivative of the Heaviside function H in the normal direction \vec{n} is called Dirac delta function, denoted $\hat{\delta}(x)$:*

$$\hat{\delta}(x) = H(\phi)' \cdot \vec{n} = H'(\phi) \nabla \phi \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = H'(\phi) |\nabla \phi| = \delta(\phi) |\nabla \phi|.$$

With two-dimensional segmentation in mind we consider \mathbb{R}^2 , and can determine important properties of the closed curve as follows. Simple examples include the area of the interior region Ω_1 , and exterior region Ω_2 , given by

$$\int_{\Omega} H(\phi) dx, \quad \text{and} \quad \int_{\Omega} (1 - H(\phi)) dx,$$

respectively. The length of the interface $\partial\Omega_1$ is

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

In the next section we will discuss how these ideas are incorporated into an alternative method to curve parametrisation.

2.6.3 Level Set Method

The level set method is a numerical technique to track a moving interface or surface. First introduced by Osher and Sethian [95] in 1988, it has become an essential method in many applications in computer vision and image processing. In relation to our work it is noteworthy as it provided a formulation to efficiently track an evolving contour, and is central to many important models [33, 34, 123]. In this section we provide a brief overview of this work. We refer the reader to the original paper [95] and other work [143, 83] for further detail. Whilst it is applicable in higher dimensions (i.e. tracking surfaces) we concentrate on curve (or interface) evolution, as it is relevant to two-dimensional imaging.

For a given interface $\Gamma \in \Omega$, the level set method consists of implicitly representing Γ with the zero level set of a Lipschitz function $\phi : \Omega \rightarrow \mathbb{R}$, such that

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

This is illustrated in Fig. 2.1, where the interface for a corresponding level set function is given.

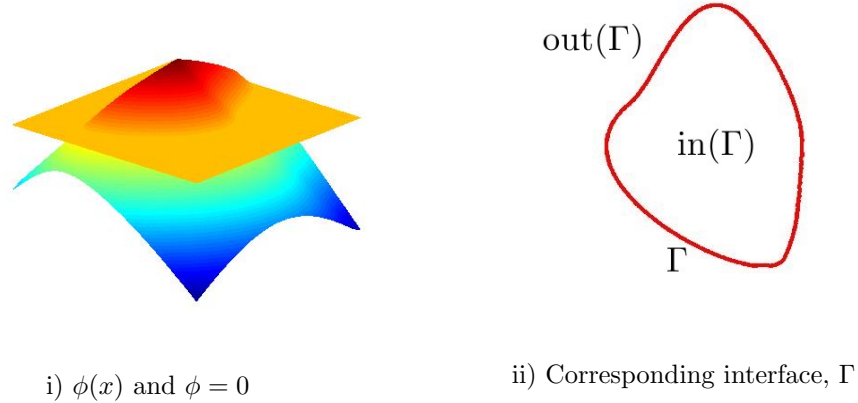


Figure 2.1: Illustration of the interface representation with the level set method. i) shows a function ϕ (conventionally a distance function) and its intersection with $\phi = 0$. ii) shows the corresponding zero level set of ϕ , which implicitly defines the interface based on the values of ϕ . The level set function is almost arbitrary (excepting possible numerically difficult choices) in this context as long as Γ remains unchanged.

The interface evolves over time so it is necessary to define the level set as a function of time:

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

Now, consider an evolving interface where the normal velocity $v(x)$ is known for every point x with $\phi(x) = 0$. Then, the evolution can be found by solving the following ordinary differential equation:

$$\begin{cases} \frac{dx(t)}{dt} = v(x) \\ \Gamma(t = 0) = \Gamma_0, \end{cases}$$

for every point x on the interface Γ . This is the Lagrangian formulation of the interface evolution equation. In order to overcome the difficulties associated with parametrising this interface, Osher and Sethian proposed the level set formulation. In relation to the ordinary differential equation above, the solution can be given by solving the following PDE:

$$\begin{cases} \frac{\partial \phi(x)}{\partial t} = -F(x)|\nabla \phi(x)| \\ \phi(x, t = 0) = d(\Gamma_0) = \phi_0(x), \end{cases}$$

where $d(\cdot)$ is generally a signed distance function, whose zero level set is the initial contour Γ_0 , and F is the speed in the outward normal direction, i.e. $v(x) = F(x(t)) \cdot \vec{n}$. This alternate method of tracking interface evolution is much more effective than a parametrisation approach, and can be applied in practice by selecting F accordingly. For example, in imaging it is natural to consider a term based on an edge detector of an observed image such that the interface defines object boundaries.

From the above we can make some observations about features of the level set method, and why it has emerged as a useful tool for image segmentation. One of the main difficulties with parametrising the interface Γ is that it relies on the object in question being simple topologically. By defining the interface implicitly, in a parametrisation free formulation, topological changes of Γ are dealt with automatically. As the function ϕ evolves, the interface Γ can split or merge without difficulty, as it is defined by the zero level set of ϕ . With an explicit parametrisation, this would be challenging. In the case of splitting, the algorithm would have to construct separate parametrisations that would then evolve further and potentially split or merge in other areas of the domain. Also, the only points of ϕ of interest are those on the zero level set, i.e. the interface. This means that the level set function is essentially arbitrary, as long as the interface is consistent. This allows for adjustments to be made to ϕ for the purposes of numerical stability, known as reinitialisation. We refer the reader to the work of Li et al. [83] and the references therein, for further details.

2.7 Iterative Solutions to Equations

In this section we discuss how the equations that arise in later chapters are solved. These are split into two classes, which require different approaches. The first is linear systems, which are relatively straightforward to solve depending on the structure of the system. We introduce fundamental methods such as the Jacobi Method, and the Gauss Seidel Method. Other examples similar to these include Successive Over Relaxation [94], which we do not address here. This addresses the principle of iterative methods with some initial approximation, that approaches the true solution. The second is nonlinear equations. We introduce Newton's Method, gradient descent and time marching schemes, and finally additive operator splitting (AOS) [127, 128, 129, 85], which splits an m -dimensional problem into a series of m one-dimensional problems, and is related to a semi-implicit time marching scheme. These methods are more commonly necessary for the equations that arise in variational imaging problems, such that they are essential to consider.

2.7.1 Basic Methods for Linear Systems

First, we introduce and review some basic iterative methods for finding solutions to linear systems of equations of the form

$$Au = b, \tag{2.14}$$

where A is an $n \times n$ matrix, b is an $n \times 1$ vector and u is the $n \times 1$ vector of unknowns. Finding the unknown u with direct methods, such as Gaussian elimination, requires the calculation of the inverse of A such that we have a solution $u = A^{-1}b \in R^N$. Numerically, direct methods are not suitable for many applications as they have substantial computational costs. The methods introduced in the following use some initial approximation $u^{(0)}$ to generate a sequence of approximations that approach the true solution u . Such methods involve iterations of the form

$$u^{(k)} = Tu^{(k-1)} + c,$$

where k is the current iterative step, T is an update matrix, and c some vector, neither of which is dependent on the iterative sequence. How these are defined depends on the technique used, which we will now address. They have an advantage over alternatives in terms of implementation and their cheap computation, such that they are an important method to consider.

The Jacobi Method

The Jacobi method, named after Carl Gustav Jakob Jacobi, is a simple iterative scheme that forms the basis of many other methods. For the original system of linear equations (2.14), we can see that the i th equation is given by

$$\sum_{j=1}^n a_{ij}u_j = b_i. \quad (2.15)$$

Solving for u_i is then given by the equation

$$u_i = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{a_{ij}u_j}{a_{ii}} \right).$$

Generalising for the k th update, assuming that the previous iterations $u^{(1)}, \dots, u^{(k-1)}$ have been calculated, we then have the Jacobi update:

$$u_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}u_j^{(k-1)} \right), \quad \text{for } i = 1, \dots, n.$$

This method is consistent with a parallel implementation, which can provide significant gains in speed. The next method modifies this idea in the sense that it uses the most recent approximation of x to improve performance.

Gauss Seidel Method

The Gauss Seidel method, named after Carl Friedrich Gauss and Philipp von Seidel, is closely related to the Jacobi method. Essentially, the method consists of Jacobi iterations using the most recent values at each iteration. In particular, when calculating $u_i^{(k)}$ the values $u_1^{(k)}, \dots, u_{i-1}^{(k)}$ are used. The idea is that these are better approximations

of the solution than $u_1^{(k-1)}, \dots, u_{i-1}^{(k-1)}$, and so can provide some improvement. Gauss Seidel iterations are defined as follows:

$$u_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} u_j^{(k)} - \sum_{j=i+1}^n a_{ij} u_j^{(k-1)} \right) \quad \text{for } i = 1, \dots, n.$$

Each update is very dependent on previously updated entries, meaning the ordering of the equations is vital. Convergence is quicker when a_{ii} is as large as possible. An important advantage of Gauss Seidel is that only one storage vector is required, as opposed to Jacobi which requires two arrays $u^{(k-1)}$ and $u^{(k)}$. Instead, each entry is replaced as soon as it is not required. For large systems of equations this is particularly beneficial, as the memory required can be significantly reduced. As with Jacobi, this method can be implemented in parallel to speed up computation.

Convergence

Each of the iterative methods above define a sequence of the form

$$u^{(k)} = T u^{(k-1)} + c,$$

for a particular iteration matrix, T . In order to discuss some basic concepts about the convergence of such schemes, we introduce some relevant definitions.

Definition 2.7.1 (Symmetric Matrix). *A square matrix A is called symmetric if it is equal to its transpose, A^T . That is, $A = A^T$.*

Definition 2.7.2 (Positive Definitive Matrix). *A real symmetric matrix A is positive definite if $u^T A u > 0$ for all non-zero vectors u . This is equivalent to saying that all the eigenvalues of A are positive.*

Remark 2.7.3 *If the matrix A is positive definite then it can be shown that its inverse, A^{-1} , exists [94].*

Definition 2.7.4 (Diagonally Dominant Matrix). *A matrix A is said to be diagonally dominant if for each row the absolute value of the entry on the diagonal is greater than or equal to the sum of the absolute values of the entries off the diagonal. Precisely:*

$$|a_{ii}| \geq \sum_{i \neq j} |a_{ij}|, \quad \forall i.$$

A matrix A is called strictly diagonally dominant if

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}|, \quad \forall i.$$

Remark 2.7.5 *If A is strictly diagonally dominant then it is not singular, meaning the original system $Au = b$ has a unique solution. In this case, both methods always converge and Gauss Seidel is faster than Jacobi iterations. It can also be shown that Gauss Seidel will converge for any $u^{(0)}$ [94].*

Generally speaking, variational formulations in imaging rarely lead to equations that can be solved with the above methods. Instead a nonlinear system is produced, and we will introduce some basic methods for that case next.

2.7.2 Nonlinear Equations

We now introduce the problem of finding solutions to systems of nonlinear equations. Generally, the equations that arise in variational imaging problems are nonlinear and as such considering these methods is important. In this short section we introduce some basic methods to solve nonlinear equations of this type. We begin by introducing descent methods for minimising an energy functional and then discuss the AOS method, which is applicable to a certain class of nonlinear equations of interest. These approaches have been used for some problems associated with imaging [105, 104, 9]. Finally, we provide a brief review of Newton's Method which is referred to in a later chapter in relation to affine registration. However, recently there have been significant developments in algorithms associated with solving nonlinear equations derived from minimising imaging formulations. We address this topic in Chapter 3 where we briefly discuss the state-of-the-art in this area. However, in this section we address nonlinear methods in general.

Gradient Descent Method

Descent methods are a common approach to computing a minimiser of nonlinear functionals. Let $F : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable function. Descent methods are similar in nature to Newton's method, requiring an initial estimate $u^{(0)} \in \mathbb{R}^n$, and an iteration scheme:

$$u^{(k)} = u^{(k-1)} - \alpha^{(k-1)} s^{(k-1)}, \quad k = 1, 2, \dots \quad (2.16)$$

Here $s^{(k-1)}$ is a search direction (dependent on the choice of method), and $\alpha^{(k-1)} > 0$ is the step length. This procedure aims to successively move closer to the true solution, u^* . A particular case of the descent method is gradient descent, where the search direction is opposite to the gradient of F , $\nabla F(u^{(k-1)})$. The idea is that the function F decreases fastest in this direction, and hence this method is also known as Steepest Descent. Therefore, the gradient descent scheme is given as follows:

$$u^{(k)} = u^{(k-1)} - \alpha^{(k-1)} \nabla F(u^{(k-1)}), \quad k = 1, 2, \dots \quad (2.17)$$

The main characteristic of descent methods is that the iteration scheme reduces the value of the function for each k :

$$F(u^{(k)}) \leq F(u^{(k-1)}). \quad (2.18)$$

This condition is satisfied by selecting the step length $\alpha^{(k-1)}$ appropriately, i.e. sufficiently small.

A particular case of the gradient descent method is that of time marching. This is when the step length is fixed for some time step, τ . This method is restricted in the sense that the stability of the scheme is heavily dependent on choosing a small τ which increases the number of iterations required for convergence to a steady state solution, and hence $\nabla F(u) = 0$. The explicit scheme time marching scheme is given as follows:

$$u^{(k)} = u^{(k-1)} - \tau \nabla F(u^{(k-1)}), \quad k = 1, 2, \dots \quad (2.19)$$

Despite its drawbacks in computational performance, its reliability and ease of implementation has made time marching very popular. Many seminal approaches in imaging have employed time marching to obtain a solution, such as Rudin, Osher, and Fatemi [109] and Chan, Esedoglu, and Nikolova [30]. It is possible to reduce the stability restrictions on τ by employing a semi-implicit scheme, i.e. the gradient is dependent on the current approximation of the solution: $\nabla F(u^{(k)}, u^{(k-1)})$. This means that at each iteration a system of equations has to be solved to obtain $u^{(k)}$, which can possibly be more problematic depending on the equation. In the following we examine such a scheme for a nonlinear diffusion equation.

Additive Operator Splitting Scheme

Additive Operator Splitting (AOS) was first introduced by Tai et al. [85] in 1992 and Weickert et al. [129] in 1998 as an m -dimensional semi-implicit scheme, based on a discrete nonlinear diffusion scale-space framework. With a diffusivity function, $W(u)$, and reaction term, f , the diffusion equation is given as follows

$$u_t = \operatorname{div}(W(u)\nabla u) + f(x) = (W(u)u_{x_1})_{x_1} + \dots + (W(u)u_{x_m})_{x_m} + f(x), \quad (2.20)$$

in $[0, T] \times \Omega \subset \mathbb{R}^m$, and with initial and boundary conditions

$$u(0, \cdot) = u_0 \quad \text{and} \quad \frac{\partial u}{\partial n} = 0 \quad \text{on} \quad \partial\Omega,$$

where n denotes the normal to the boundaries $\partial\Omega$. It is important to note that the diffusivity function, W , is dependent on u . This is crucial as it relates to the PDEs arising from imaging models, such as ROF or GCS, by partially lagging the mean curvature term. We will return to this later. A discrete m -dimensional function $u(x)$ can be considered a vector where element i represents the location x_i . We consider discrete times $t_k := k\tau$ ($k \in N_0$), and τ is the time step size. We denote the approximations to $u(x_i, t_k)$ and $W(u(x_i, t_k))$ by u_i^k and W_i^k respectively. For comparison, we first consider the conventional semi-implicit m -dimensional scheme. With a backward Euler implicit step for the time discretisation and a spatial finite difference scheme, a semi-implicit discretisation of the diffusion equation with reflecting boundary conditions is given by

$$\begin{aligned} \frac{u^{k+1} - u^k}{\tau} &= A(u^k)u^{k+1} + f \\ (I - \tau A(u^k)u^{k+1} &= u^k + \tau f \\ u^{k+1} &= \left(I - \tau A(u^k)\right)^{-1} (u^k + \tau f), \quad k = 1, 2, \dots \end{aligned} \quad (2.21)$$

where u^k is represented by a column vector of length N^m . For dimensions $m \geq 2$, the matrix $A(u^k)$ will have a large bandwidth and applying direct algorithms lead to

computational storage difficulties, or slow convergence due to large time steps increasing the condition number of the system matrix [129] in the case of classical iterative algorithms.

The AOS scheme is a splitting-based alternative method where the one-dimensional Thomas algorithm can be used [129] m times at each iteration. With this in mind, a discrete version of (2.20) is given by

$$\frac{\partial u}{\partial t} = \sum_{j=1}^m \frac{\partial}{\partial x_i} (W_j(u) \frac{\partial u}{\partial x_j}) + f \quad (2.22)$$

In one spatial dimension, the semi-implicit scheme can be given as

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \sum_{j \in \mathcal{N}_l(i)} \frac{W_j^k + W_i^k}{2h^2} (u_j^{k+1} - u_i^{k+1}) + f, \quad (2.23)$$

where $\mathcal{N}_l(i)$ consists of the two neighbours of element i along the l direction (boundary elements may have only one neighbour). The AOS scheme treats each direction separately to exploit the advantages of solving one-dimensional problems. It is given as follows:

$$u^{k+1} = \frac{1}{m} \sum_{l=1}^m \left(I - m\tau A_l(u^k) \right)^{-1} (u^k + \tau f), \quad k = 1, 2, \dots \quad (2.24)$$

where the operators $B_l(u^k) := I - m\tau A_l(u^k)$ describe one-dimensional diffusion processes along the x_l axes. Each iteration step requires the previous iterate to be propagated in each coordinate direction separately. The new iteration is then given by the average of these intermediate solutions. For example, consider the case of most interest for our problems (i.e. $m = 2$). The matrices A_ℓ are the diffusion quantity in the ℓ direction ($\ell = 1, 2$ for x and y directions respectively) and are given as follows, where h_l denotes the grid size:

$$\begin{aligned} \left(A_1(u^k) u^{k+1} \right)_{i,j} &= \left(\partial_x (W(u^k) \partial_x u^{k+1}) \right)_{i,j} \\ &= \frac{1}{h_1} \left(W_{i+1/2,j}^k (\partial_x u^{k+1})_{i+1/2,j} - W_{i-1/2,j}^k (\partial_x u^{k+1})_{i-1/2,j} \right) \\ &= \frac{1}{h_1} \left(\frac{W_{i+1,j}^k + W_{i,j}^k}{2} \frac{u_{i+1,j}^{k+1} - u_{i,j}^{k+1}}{h_1} - \frac{W_{i,j}^k + W_{i-1,j}^k}{2} \frac{u_{i,j}^{k+1} - u_{i-1,j}^{k+1}}{h_1} \right) \\ &= u_{i+1,j}^{k+1} \left(\frac{W_{i+1,j}^k + W_{i,j}^k}{2h_1^2} \right) + u_{i-1,j}^{k+1} \left(\frac{W_{i-1,j}^k + W_{i,j}^k}{2h_1^2} \right) \\ &\quad - u_{i,j}^{k+1} \left(\frac{W_{i+1,j}^k + W_{i-1,j}^k + 2W_{i,j}^k}{2h_1^2} \right) \end{aligned}$$

and similarly,

$$\begin{aligned}
\left(A_2(u^k)u^{k+1}\right)_{i,j} &= \left(\partial_y(W(u^k)\partial_y u^{k+1})\right)_{i,j} \\
&= u_{i,j+1}^{k+1} \left(\frac{W_{i,j+1}^k + W_{i,j}^k}{2h_2^2}\right) + u_{i,j-1}^{k+1} \left(\frac{W_{i,j-1}^k + W_{i,j}^k}{2h_2^2}\right) \\
&\quad - u_{i,j}^{k+1} \left(\frac{W_{i,j+1}^k + W_{i,j-1}^k + 2W_{i,j}^k}{2h_2^2}\right).
\end{aligned}$$

AOS is a semi-implicit scheme that requires m tridiagonal linear systems to be solved at each iteration. Due to the structure of the discretisation of the operator the Thomas algorithm can be used. For such systems, the solution can be obtained in $\mathcal{O}(n)$ operations instead of $\mathcal{O}(n^3)$ required by Gaussian elimination. To demonstrate the reliability of the AOS scheme we will now briefly discuss the criteria for nonlinear diffusion scale-spaces and the advantages of satisfying such conditions [127, 128]. Without loss of generality we drop the reaction term f and we have. For a given discrete scheme of type

$$u^0 = f \tag{2.25}$$

$$u^{k+1} = Q(u^k)u^k, \quad \forall k \in \mathbb{N}_0 \tag{2.26}$$

the following criteria must hold:

(D1) *Continuity in its argument:*

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

(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 $k_0, \dots, k_r \in J$ with $k_0 = i$, and $k_r = j$ such that $q_{k_p k_{p+1}} \neq 0$ for $p = 0, \dots, r-1$.

In Weickert et al. [129], the above criteria are demonstrated to have fulfilled a discrete

scale space. This is important for a number of reasons, which we will not detail in full here. In particular, there is convergence to a constant steady state:

$$\lim_{k \rightarrow \infty} u_i^k = \mu, \quad \forall i \in J. \quad (2.27)$$

Additionally, there are no restrictions on the time step size as the scheme is unconditionally stable. This demonstrates clear advantages of AOS over explicit and semi-implicit time marching schemes. This is especially true when $m > 2$. This scheme forms the basis for a proposed scheme in Chapter 4, where we apply this method to GCS.

Remark 2.7.6 *The AOS scheme (2.24) with $m=2$ corresponding to the finite difference equation*

$$(\mathcal{F}u)^k := \frac{1}{\tau}u^{k+1} - \frac{1}{2\tau}\left(I - 2\frac{\tau}{h^2}A_1\right)^{-1}u^k - \frac{1}{2\tau}\left(I - 2\frac{\tau}{h^2}A_2\right)^{-1}u^k = 0,$$

$k = 0, 1, \dots$, is consistent in l_∞ -norm of first order in time and second order in space with the PDE (2.22).

Newton's Method

A typical representation of a nonlinear system of equations is $F(u) = v$, where $v(x) \in \mathbb{R}^n$ is fixed, $u(x) \in \mathbb{R}^n$ is unknown, and $F : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a continuously differentiable, nonlinear operator. This can be rewritten in the form

$$F(u) = \mathbf{0}, \quad (2.28)$$

where $\mathbf{0}$ represents the zero vector. The problem consists of obtaining a solution to (2.28), $u^* \in \mathbb{R}^n$. Let J denote the Jacobian matrix $(\partial F_i / \partial x_j)$ of F , and assume that J is Lipschitz continuous (see Definition 2.1.13). The method consists of beginning with an initial approximation $u^{(0)}$ and carry out the following iterations:

$$u^{(k)} = u^{(k-1)} - J\left(u^{(k-1)}\right)^{-1} F\left(u^{(k-1)}\right), \quad k = 1, 2, \dots$$

With each iteration the aim is to find approximations that are closer to the solution u^* . However, computing the inverse of the Jacobian can be avoided by rearranging the equation as follows:

$$J\left(u^{(k-1)}\right)r^{(k)} = -F\left(u^{(k-1)}\right).$$

Then, a linear system has to be solved to give $r^{(k)} = u^{(k)} - u^{(k-1)}$. Then, the iterate can be given explicitly:

$$u^{(k)} = u^{(k-1)} - r^{(k)}.$$

For Newton's Method the Jacobian has to be calculated at each step, meaning that it has a high computational cost. If $J\left(u^{(k-1)}\right)$ is ill conditioned obtaining $r^{(k)}$ can be problematic as well. However, if the Jacobian is non-singular at the solution, local

quadratic convergence can be proven [64]. The method is also heavily dependent on the initial estimate. If $u^{(0)}$ is close enough to u^* then Newton's Method can offer fast convergence [94].

Chapter 3

Review of Variational Methods for Imaging Processing

3.1 Introduction

In this chapter we briefly review variational methods for image segmentation, introducing relevant models in relation to our work. Further background can be found in the work of Mitiche and Ben Ayed [87]. In relation to image processing, variational methods involve treating the observed discrete image in the continuous domain, and minimising a functional that leads to a corresponding equation. The solution of this equation relates to the original image processing problem. Variational methods are closely related to stochastic approaches, and many of the problems discussed below can be formulated in an analogous way using Bayesian maximum a posteriori estimation. However, in this work we concentrate on the variational approach which we will define more specifically in the following. For the link between variational and stochastic methods, the work of Chan and Shen [31] can provide an overview.

Variational methods consist of minimising energy functionals that define some constraints on the objective function. Let us define the solution as

$$u^* = \arg \min_{u \in S} F(u),$$

where u^* is an optimiser of the functional $F(\cdot)$, defined on an appropriate space S . If F is continuous and differentiable, the first variation can be computed leading to the Euler-Lagrange equation

$$\frac{\partial F}{\partial u} = 0, \tag{3.1}$$

which gives a necessary condition for u^* to be an optimiser of F such that

$$\left. \frac{\delta F}{\delta u} \right|_{u^*} = 0.$$

Typically, a functional is formulated in the following way

$$F(u) = \underbrace{\alpha \int_{\Omega} \mathcal{J}(u) dx}_{\text{Regularisation}} + \underbrace{\int_{\Omega} f(u) dx}_{\text{Fitting}},$$

where f is a fitting function that stipulates the correspondence between the objective function and the data, and \mathcal{J} is a term that imposes regularity. These terms are discussed in the context of inverse problems in Section 2.3, and further details can be found in the literature [87, 32, 121, 124].

In this chapter we address some methods of particular interest to our work, and important to the subject in general. In Section 3.1.1 we discuss image denoising. This is a fundamental problem in image processing, and we discuss it in the context of Tikhonov regularisation in Example 2.2.8. A seminal approach to this problem was introduced by Rudin, Osher, and Fatemi (ROF) [109] with total variation (TV) regularisation. This is important to our work, as it is closely related to conventional segmentation methods that also involve TV terms [33, 30]. It is also relevant to the problem of image deblurring (or deconvolution), which we introduce in Section 3.1.2. This is directly related to our work in the sense that Chapter 7 addresses the problem of segmentation for an image degraded with unknown blur. We discuss related work such as Chan and Wong [35] and You and Kaveh [135]. We then introduce the idea of image registration in Section 3.1.3. We briefly mention important variational approaches to introduce this important concept in image processing, but detail a parametric method for affine registration. This idea has relevance to our work in Chapter 8, where we incorporate shape priors for variational segmentation.

In Section 3.2 we discuss the main focus of our work, in reviewing prominent variational segmentation methods in the literature. We begin with the work of Mumford and Shah [89], which was first introduced in 1989. This work is closely linked to the seminal work of Geman and Geman [55] in the discrete setting. We also consider a particular case of this formulation, based on a piecewise-constant framework discussed by Chan and Vese [33]. The most important aspect of this work in relation to this thesis is that of convex relaxation methods, which we introduce in Section 3.2.4. Here we introduce the framework of two-phase variational segmentation problems for which one can compute global minimisers, which is the main subject of our work. We expand on this in Chapter 4 and more detail can be found in the literature [17, 29, 30, 18, 25, 78]. Finally, we offer some brief remarks on current algorithms applicable to imaging. Of particular interest is Chambolle’s dual formulation [23], which was first applied to segmentation by Bresson et al. [18] and will be referred to again in Chapters 4 and 6.

3.1.1 Denoising

Noise is present in almost any image, often introduced in the acquisition or storage of data. There are many different types of noise, such as Gaussian which is additive and independent of the signal intensity, and Poisson which is dependent on scene brightness. Further details about types of noise can be found in [32], and understanding this is essential to modeling its removal. In the variational framework, the seminal work of ROF [109] was introduced in 1992. It is important for its ability to remove noise whilst preserving edges, as opposed to previous approaches that employed L^2 regularisation of the gradient (given in Example 2.3.2). This is achieved by employing TV regularisation,

which has been detailed in the previous chapter. The formulation is given as follows:

$$\min_u \left\{ \alpha \int_{\Omega} |\nabla u| \, dx + \frac{1}{2} \int_{\Omega} |u - z|^2 \, dx \right\} \quad (3.2)$$

where α is a positive parameter controlling the level of noise removal. It is important to note that (3.2) is well-posed, and thus we can guarantee the existence and uniqueness of the minimiser (the functional is strictly convex.) The Euler-Lagrange equation is given formally as follows:

$$u - z - \alpha \operatorname{div} \left(\frac{Du}{|Du|} \right) = 0. \quad (3.3)$$

However, it is important to consider cases where $Du = 0$. This is especially important when considering $z \in L^\infty(\Omega)$, and it is quite possible that u will exhibit staircasing properties. With this in mind, we now derive the Euler-Lagrange equation with respect to convex analysis. Necessary details are given in Chapter 2, Ekeland and T  mam [51], and we follow the work of Chambolle et al. [24]. We first introduce the set

$$\mathcal{K} = \{-\operatorname{div} \phi : \phi \in C_0^\infty(\Omega; \mathbb{R}^n) : |\phi(x)| \leq 1 \, \forall x \in \Omega\}$$

and the closure K of \mathcal{K} in $L^2(\Omega)$, which is

$$K = \{-\operatorname{div} h : h \in L^\infty(\Omega; \mathbb{R}^n) : -\operatorname{div}(h\chi_\Omega) \in L^2(\mathbb{R}^n) : |h(x)|_\infty \leq 1 \, \forall x \in \Omega\}.$$

Let us denote, following the definition in Chapter 2, J as

$$J(u) = TV(u) = \sup_{p \in \mathcal{K}} \int_{\Omega} u(x)p(x) \, dx.$$

If $u \in L^2(\Omega)$, then

$$J(u) = \sup_{p \in K} \int_{\Omega} u(x)p(x) \, dx. \quad (3.4)$$

This leads to,

$$K = \left\{ p \in L^2(\Omega) : \int_{\Omega} p(x)u(x) \, dx \leq J(u) \, \forall u \in L^2(\Omega) \right\},$$

i.e. K is the largest set in $L^2(\Omega)$ such that (3.4) holds for any $u \in L^2(\Omega)$. Therefore, according to [24], for $u \in L^2(\Omega)$:

$$\partial J(u) = \left\{ p \in K : \int_{\Omega} p(x)u(x) \, dx = J(u) \right\}.$$

We can now derive the equation satisfied by u that minimises ROF (3.2), i.e. for any $v \in L^2(\Omega)$ we have

$$\alpha J(v) + \frac{1}{2} \int_{\Omega} (v - z)^2 \, dx \geq \alpha J(u) + \frac{1}{2} \int_{\Omega} (u - z)^2 \, dx. \quad (3.5)$$

It is then straightforward to show that

$$\alpha J(v) \geq \alpha J(u) + \int_{\Omega} (v - u)(z - u) dx - \frac{1}{2} \int_{\Omega} (u - v)^2 dx.$$

Then, for any $t \in \mathbb{R}$, according to [24]:

$$\alpha(J(u + t(v - u)) - J(u)) - t \int_{\Omega} (v - u)(z - u) dx \geq 0.$$

This demonstrates that

$$\frac{z - u}{\alpha} \in \partial J(u).$$

It follows that the Euler-Lagrange equation for the ROF model is

$$u - z + \alpha \partial J(u) \ni 0. \tag{3.6}$$

Further details can be found in the work of Chambolle et al. [24], and related literature. In practice, we look for a solution in a discretised setting where the problem is regularised and therefore considering subdifferentials is not necessary. In the rest of the thesis, we do not consider the formal definition of the Euler-Lagrange equation for a BV function as described above. We detail how we handle these equations numerically in subsequent chapters. The Euler-Lagrange equation, as stated in [109], is given as follows:

$$\begin{aligned} u - z - \alpha \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) &= 0 \quad \text{in } \Omega, \\ \nabla u \cdot \vec{n} &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{3.7}$$

with \vec{n} the unit outward normal. In ROF they use time marching to find a solution, although many methods have been applied over the years to obtain a solution [26, 59]. Despite its success and continued importance in the literature, the results demonstrate the so-called staircase effect where smooth regions are reconstructed as piecewise-constant and thus limiting quality.

This work has attracted a lot of attention, particularly with respect to reducing the staircase effect. In 2010, Bredies et al. [16] introduced Total Generalized Variation. Here, they propose a regularisation functional with derivatives of order up to k of the desired object and demonstrates improvements over lower order methods. The authors state that it should advance a wider class of problems, beyond denoising.

3.1.2 Deblurring

It is important to consider another type of image degradation. Whereas the process of denoising is incorporated into many current segmentation methods [33, 30], when an image is blurred it is often beyond conventional segmentation methods. Blurring of images is common in many areas, such as astronomical or medical imaging, such as colour fundus angiography for retinal imaging. The process of image deblurring (or

deconvolution) is the task of reconstructing the true image from the observed degraded image, restoring important features. We consider the problem of segmenting blurred images in Chapter 7, and therefore introduce some ideas relating to this problem and summarise a seminal work [35] in the following.

Mathematically, convolution is an operation on two functions f and g , producing a function that is considered the amount of overlap of one function as it is shifted over the another function. In image processing a kernel consists of an image operator changing the value of the image pixels x depending on the neighborhoods pixel values.

Definition 3.1.1 *Let $f(t)$ and $w(t)$ be two functions. The convolution of f and w , denoted by $w * f$, is the function on $t \geq 0$ given by*

$$g(x) = w * f(x) = \int_0^x w(s)f(x - s) ds.$$

This is a particular kind of integral transform:

$$w * f(x) = \int_{-\infty}^{\infty} w(s) f(x - s) ds, \quad (3.8)$$

or more generally, if f and w are complex-valued functions on \mathbb{R}^n :

$$w * f(x) = \int_{\mathbb{R}^n} w(s) f(x - s) ds. \quad (3.9)$$

Since the image is stored as a collection of discrete pixels we need a discrete convolution form. In discrete form the integral (3.9) is replaced by summation, for example for a 2-D function integral (3.9) can written as:

$$g(x, y) = w(x, y) * f(x, y) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w(s, t) f(x - s, y - t) ds.$$

An observed blurred image can be written as a convolution of the true image with a point spread function or unknown kernel function. Letting $z(x)$ denote the received image, $h(x)$ the blur function, $\eta(x)$ the noise acquired during data collection, and $u(x)$ the true image to be recovered. We model the blurred image as

$$z(x) = [h * u](x) + \eta(x).$$

There are three main deblurring problems.

1. *Non-blind deconvolution*: The point spread function is assumed known, despite this being rare in real applications. The aim is to recover the true image, and has been widely studied.
2. *Semi-blind deconvolution*: Information about the blur kernel is assumed or known. The task is to estimate the correct blur parameters [12].
3. *Blind image deconvolution*: The image and blur kernel are both unknown, and the task of recovering both is very challenging. [35, 38].

I will briefly review a prominent model that tackles the third problem mentioned above: Blind Image Deconvolution. This model was introduced by Chan and Wong [35] in 1998, as an extension of a model by You and Kaveh [135]. The Chan-Wong functional is given as

$$f(u, h) = \frac{1}{2} \|h(x) * u(x) - z(x)\|^2 + \alpha_1 \int_{\Omega} |\nabla u(x)|_{\beta} + \alpha_2 \int_{\Omega} |\nabla h(x)|_{\beta},$$

where α_1 and α_2 are small, non-negative parameters. The regularisation on u and h is a smooth approximation of the TV function:

$$\int_{\Omega} |\nabla h(x)|_{\beta} = \int_{\Omega} \sqrt{|\nabla h(x)|^2 + \beta^2}.$$

The functional $f(u, h)$ is not jointly convex with respect to u and h . Accordingly, it is minimised subject to the following constraints [35, 135]:

$$u(x) \geq 0, \quad h(x) \geq 0, \quad \int_{\Omega} h(x) dx = 1, \quad h(x) = h(-x). \quad (3.10)$$

To minimise the functional the Euler-Lagrange equations are derived:

$$\begin{aligned} \frac{\partial}{\partial u} f(u, h) : \quad h^{\dagger}(x) * (h(x) * u(x) - z(x)) + \alpha_1 \nabla \cdot \left(\frac{u(x)}{|u(x)|_{\beta}} \right) &= 0, \\ \frac{\partial}{\partial h} f(u, h) : \quad u^{\dagger}(x) * (u(x) * h(x) - z(x)) + \alpha_2 \nabla \cdot \left(\frac{h(x)}{|h(x)|_{\beta}} \right) &= 0, \end{aligned}$$

where $h^{\dagger}(x) = h(-x)$ and $u^{\dagger}(x) = u(-x)$ are the adjoints of h and u , respectively. A scheme of alternate minimisation is described where the constraints introduced above are imposed at each stage. Further details are found in [35]. We will cover these ideas more thoroughly in Chapter 7, and introduce some models that address the task of joint segmentation and deblurring.

3.1.3 Registration

Image registration is the challenging task of aligning two images to establish a correspondence between the features within them. Modersitzki [88] classifies intensity based registration methods by two definitions: non-parametric and parametric. The former is based on the variational approach, where the aim is to find a deformation field based on physical processes such as curvature. Examples include Ibrahim et al. [66] and [138], where the dissimilarity functionals are based on intensity difference in the whole image. Alternative approaches involve landmark based methods, such as Lui et al. [71], where certain locations define the deformation field. It is possible to incorporate non-linear registration methods into variational segmentation models, such as Cremers et al. [39], but typically these approaches involve parametric methods (i.e. not variational based). We will cover these methods in detail in Chapter 8, but some examples include [130, 36]. Parametric image registration involves a rigid or affine transformation, which are dependent on three and six parameters, respectively. As this method is of particular interest we briefly discuss some details related to affine registration.

The template and reference, $T, R \in \Omega$, define the images to be compared. For $x \in \Omega$ denote by $\phi(x) : \Omega \rightarrow \Omega$ the unknown coordinate transformation that produces the alignment between the reference, $R(x)$ and the transformed template, $T(\phi(x))$. We address the problem where we assume the target object is approximately an affine transformation of the shape prior, such that the segmentation closely favours shapes given by the prior information. This means the transformation is linear and can be defined as follows:

$$\phi(x) = \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \end{bmatrix}, \quad (3.11)$$

given six parameters $\mathbf{a} = \{a_1, a_2, a_3, a_4, a_5, a_6\}$. The regularised affine registration model [41] is as follows:

$$\min_{\mathbf{a}} \left\{ \eta \mathcal{R}(\mathbf{a}) + \frac{1}{2} \int_{\Omega} \left(T(\phi(x)) - R(x) \right)^2 dx \right\},$$

where $\eta > 0$ is a weighting parameter for the regularisation of \mathbf{a} , and the sum of squared differences (SSD) term determines the similarity between the reference and template. In the following sections we detail how linear registration methods are incorporated into variational segmentation models. A reasonable choice of regularisation for the affine parameters \mathbf{a} is:

$$\mathcal{R}(\mathbf{a}) = \frac{1}{2} \left((1 - a_1)^2 + a_2^2 + a_3^2 + a_4^2 + (1 - a_5)^2 + a_6^2 \right). \quad (3.12)$$

Other choices of \mathcal{R} are possible, and good results can also be attained with no regularisation for some examples. The similarity measure we use is the SSD, which we define as follows:

$$\mathcal{D}(\mathbf{a}) = \frac{1}{2} \int_{\Omega} \left(T(\phi(x)) - R(x) \right)^2 dx, \quad (3.13)$$

The minimisation of the affine registration step is then given as

$$\min_{\mathbf{a} \in \mathbb{R}^6} \left\{ \mathcal{J}_{\eta}(\mathbf{a}) = \eta \mathcal{R}(\mathbf{a}) + \mathcal{D}(\mathbf{a}) \right\}. \quad (3.14)$$

We solve this with the discretise-optimise approach, which we will briefly discuss in general terms. Using the Gauss Newton method we can update \mathbf{a} with a perturbation $\delta \mathbf{a}^{(k)}$,

$$\mathbf{a}^{(k+1)} = \mathbf{a}^{(k)} + \delta \mathbf{a}^{(k)}, \quad (3.15)$$

The Gauss Newton perturbation $\delta \mathbf{a}^{(k)}$ is given by

$$\tilde{\mathbf{H}}_{\mathcal{J}_{\eta}}(\mathbf{a}^{(k)}) \delta \mathbf{a}^{(k)} = -\mathbf{g}_{\mathcal{J}_{\eta}}(\mathbf{a}^{(k)}) \quad (3.16)$$

where

$$\tilde{\mathbf{H}}_{\mathcal{J}_{\eta}}(\mathbf{a}^{(k)}) = \mathbf{J}^{\top}(\mathbf{a}^{(k)}) \mathbf{J}(\mathbf{a}^{(k)}) + \eta \mathbf{H}_{\mathcal{R}}(\mathbf{a}^{(k)})$$

and

$$\mathbf{g}_{\mathcal{J}_\eta}(\mathbf{a}^{(k)}) = \nabla_{\mathbf{a}}\mathcal{D}(\mathbf{a}) + \eta\nabla_{\mathbf{a}}\mathcal{R}(\mathbf{a}^{(k)})$$

are the approximated Hessian and the gradient of \mathcal{J}_η at $\mathbf{a}^{(k)}$ and $\mathbf{H}_{\mathcal{R}}(\mathbf{a}^{(k)})$ are the gradient and Hessian of \mathcal{R} at $\mathbf{a}^{(k)}$ respectively. An approximated Hessian is used to deal with the nonlinearity of the problem. In this way computing higher order derivatives is avoided. Further details can be found in [41], and we will return to these ideas in Chapter 8.

3.2 Image Segmentation

Image segmentation is the partitioning of an image into multiple regions of shared characteristics. Variational methods to this end consist of minimising an energy functional, leading to a partial differential equation based on the Calculus of Variations. The solution to this equation then corresponds to a meaningful representation of the image. The aim is to find a closed contour Γ that partitions a domain $\Omega \in \mathbb{R}^2$ into subregions Ω_i , $i = 1, 2, \dots, N$. In this thesis we focus on two-phase methods. That is, we want to partition the image into some meaningful foreground/background representation, and $N = 2$. Much of the work in this thesis is based on the Mumford and Shah [89] formulation, where the aim is to find either the piecewise-smooth or piecewise-constant approximation of the image and the edge that defines the discontinuity.

Given an observed discrete image the problem setting is in the continuous domain where the aim is to determine a solution to some equation derived from an energy functional. It is unlikely that analytic solutions are available in this context, and a numerical solution of a discretised system is required. This process might seem counter-intuitive, but variational methods have proven very successful since the seminal work of Mumford and Shah. Other noteworthy developments since then include edge based methods, such as the Snakes approach of Kass, Witkin, and Terzopoulos [72] and the Geodesic Active Contours model of Caselles, Kimmel, and Sapiro [22]. Crucial to the success of these approaches was the development of level set based methods [95, 143], which have been widely used over the last twenty years. It was utilised by Chan and Vese in the influential Active Contours Without Edges [33], a region based model based on the two-phase piecewise-constant formulation of the Mumford-Shah formulation. The common theme with this approach to segmentation is that the problems are nonconvex, meaning that obtaining a global minimiser is often not possible.

Recent work addressing the issue of nonconvexity involves convex relaxation methods, essential to the work in this thesis. The foundational work in relation to segmentation in the continuous setting, is that of Chan, Esedoglu, and Nikolova [30] in 2006. This model aims to find the global minimiser of the two-phase piecewise-constant Mumford-Shah formulation, in the case of known intensity constants. The theoretical basis of this work is based on the work of Strang [119]. Related work since has included Bresson et al. [18, 78, 25, 10, 102, 137, 120]. In short, the convex relaxation method

consists of representing each phase with a binary function, $u \in \{0, 1\}$, and the interface is implicitly represented by the location of the jump. This can be seen in Fig. 3.1, and be compared to the level set based representation discussed in Chapter 2. The binary constraint is then relaxed such that $u \in [0, 1]$. We will address this in further detail in Section 3.2.4. Also of relevance to this work is Section 2.2 on functions of bounded variation and references therein. In the following sections we introduce seminal works from the subject that influence the content of this thesis.

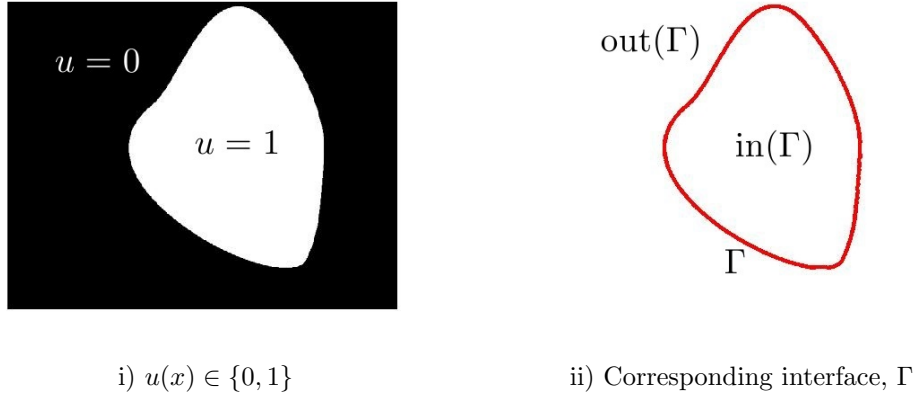


Figure 3.1: Illustration of the interface representation in the convex relaxation framework. i) shows a binary function u . ii) shows the corresponding contour, which implicitly defines the interface. In the convex relaxation framework the binary constraint is relaxed, and Γ is given by a thresholding procedure for a parameter $\gamma \in (0, 1)$.

3.2.1 Mumford-Shah Approach

An important work in this area is that of Mumford and Shah [89] in 1989. It concerns the piecewise smooth approximation of an input image $z(x)$, by a pair (u, Γ) . Let Ω be a bounded domain in \mathbb{R}^2 and $z(x)$ be a bounded measurable function defined on Ω . The Mumford-Shah functional is defined as

$$E(u, \Gamma) = \nu \mathcal{H}^{n-1}(\Gamma) + \mu^2 \int_{\Omega} (u - z)^2 dx + \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx, \quad (3.17)$$

The functional contains a fidelity term on $u \in \mathcal{C}^1$, and two regularity terms. One imposes smoothness on u , and the other imposes regularity on Γ in terms of its one-dimensional Hausdorff measure. A related functional in a discrete setting rather than on a continuous domain was first introduced by Geman and Geman [55], and studied by Blake and Zisserman [14]. Theoretical results on the existence and regularity of minimisers is discussed in [89], but computing minimisers is very challenging due to the non-regularity of the edge term Γ . A very prominent approach was introduced in 1990 by Ambrosio and Tortorelli who approximated minimisers of (3.17) by a sequence of simpler elliptic variational problems. There have been many other approaches since then that have proposed methods to find the minimal pair (u, Γ) , such as Pock et al. [100] based on convex relaxation methods, or Tsai et al. [122] and Vese et al. [34] based on level set approaches.

Mumford and Shah also discuss the restriction of E to piecewise-constant functions u . In other words, $u = c_k$ on each open set Ω_k , where the values c_k are simply the average values of z in each region Ω_k . The piecewise-constant Mumford-Shah functional is given as

$$E_0(u, \Gamma) = \nu \mathcal{H}^{n-1}(\Gamma) + \int_{\Omega_k} (u - c_k)^2 dx. \quad (3.18)$$

It can be proved that E_0 is the natural limit functional of E as $\mu \rightarrow 0$. This reduced case is also referred to as the minimal partition problem. One notes that this is linked, in the discrete setting, to the Potts Model [103] which has been widely studied [136]. We are particularly interested in the partitioning of images into foreground and background, i.e. $N = 2$. This is known as the two-phase piecewise constant Mumford-Shah functional and is given as follows:

$$E(\Gamma, c_1, c_2) = \nu |\Gamma| + \int_{\Omega_1} (z - c_1)^2 dx + \int_{\Omega_2} (z - c_2)^2 dx. \quad (3.19)$$

The above functional is the basis for a significant amount of important work in this field, and forms the basis for much of the later chapters. I will discuss it in more detail in terms of computing minimisers in relation to the Chan-Vese Model [33] and convex relaxation [30] later in this chapter.

3.2.2 Geodesic Active Contours

An early PDE-based method to extract objects in an image is the active contour model (also known as Snakes), proposed by Kass, Witkin and Terzopolous [72] in 1988. Similar approaches have been very successful and have been important in a wide range of applications, particularly in medical imaging [133, 73]. In 1997, Caselles, Kimmel, and Sapiro [22] introduced the Geodesic Active Contours model

$$F_{GAC}(C(s)) = \int_0^{L(C)} g(|\nabla z(C(s))|) ds, \quad (3.20)$$

where

$$g(\nabla z) = \frac{1}{1 + \gamma |\nabla z|^2}. \quad (3.21)$$

An example of this edge function for a given image can be seen in Fig. 3.2. The idea is that g is small near object boundaries, defined by jumps in intensities of z , and controlled by the parameter γ . The Euler-Lagrange equation of the functional and gradient descent give the following PDE:

$$\frac{\partial C}{\partial t} = g\kappa \vec{n} - (\nabla g \cdot \vec{n}) \vec{n}, \quad (3.22)$$

where κ is the Euclidean curvature, and \vec{n} is the unit normal vector. A unique viscosity solution associated with the evolution equation exists [22, 44]. As discussed in Chapter 2, an effective way to represent an interface is with the level set method, introduced by Osher and Sethian [95]. With this in mind, equation (3.22) can be rewritten as follows:

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \left(\nabla \cdot \left(g \frac{\nabla \phi}{|\nabla \phi|} \right) + \nu g \right), \quad (3.23)$$

where ϕ is a Lipschitz function representing C as a zero level set. Here the constant ν is added to attract the curves towards the boundary and increase the speed of the evolution.

Despite the success of Geodesic Active Contours it is limited in terms of applications in two senses. Firstly, the model is dependent on the gradient of the image. This means that images that either contain noise or have boundaries that are not well defined, are not suitable for this method. Secondly, and of particular interest for the purposes of this thesis, the energy functional is nonconvex and is therefore highly sensitive to initialisation.

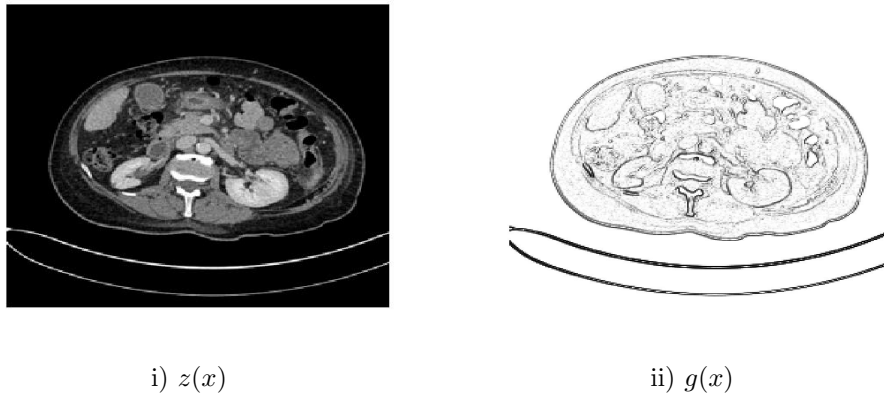


Figure 3.2: Edge detection function, $g(x)$, from eqn. (3.2) for an image, $z(x)$.

3.2.3 Active Contours Without Edges

Whilst previous active contour models relied on edge detection [22, 72], Chan and Vese introduced a two-phase region based method in 2001, known as Active Contour Without Edges [33]. The assumption behind the model is that an observed image, z , can be treated as a piecewise-constant function. Denoting Γ as the boundary partitioning the two regions of approximately constant intensity, the variational formulation is defined as follows:

$$F^{CV} = \lambda_1 \int_{\text{in}(\Gamma)} (z(x) - c_1)^2 dx + \lambda_2 \int_{\text{out}(\Gamma)} (z(x) - c_2)^2 dx + \mu \cdot \text{Length}(\Gamma) + \nu \cdot \text{Area}(\text{in}(\Gamma)), \quad (3.24)$$

where $\mu \geq 0, \nu \geq 0, \lambda_1, \lambda_2 > 0$ are fixed parameters. Generally the area constraint is ignored, i.e. $\nu = 0$, and the fitting terms are evenly balanced, i.e. $\lambda_1 = \lambda_2$. The 'Length(Γ)' term refers to the Hausdorff $(n - 1)$ -dimensional measure $\mathcal{H}^{n-1}(\Gamma)$. One notes that (3.24) is a particular case of the minimal partition problem, or the two-phase piecewise-constant Mumford Shah formulation (3.19) introduced above. To recall, the

idea is to find the best approximation, u , of the observed image z as a function taking only two values:

$$u = \begin{cases} c_1 & \text{(average } z \text{ inside } \Gamma) \\ c_2 & \text{(average } z \text{ outside } \Gamma) \end{cases}$$

Then, the minimisation problem is given as the following:

$$\min_{c_1, c_2, \Gamma} F^{CV}(c_1, c_2, \Gamma). \quad (3.25)$$

The authors use the level set method of [95, 143], introduced in brief in Chapter 2, to implicitly represent the contour Γ . This was a seminal approach with respect to image segmentation methods, as it allowed a practical and reliable numerical implementation of a particular case of the renowned Mumford-Shah formulation. We now recall some essential details of the level set method, and then reformulate (3.24). The contour Γ can be given in terms of a Lipschitz function ϕ as follows:

$$\begin{cases} \Gamma = \partial\Omega_1 = \{x \in \Omega \mid \phi(x) = 0\}, \\ \text{in}(\Gamma) = \Omega_1 = \{x \in \Omega \mid \phi(x) > 0\}, \\ \text{out}(\Gamma) = \Omega_2 = \{x \in \Omega \mid \phi(x) < 0\}. \end{cases}$$

To reformulate (3.24) we recall the definition of the Heaviside and Dirac delta function from Chapter 2:

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

Then, each term in the energy can be expressed in terms of ϕ :

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

Then, $F(\Gamma, c_1, c_2)$ can be reformulated as follows:

$$\begin{aligned} F^{LS} &= \lambda_1 \int_{\Omega} (z(x) - c_1)^2 H(\phi) dx + \lambda_2 \int_{\Omega} (z(x) - c_2)^2 (1 - H(\phi)) dx \\ &\quad + \mu \int_{\Omega} \delta(\phi) |\nabla \phi| dx + \nu \int_{\Omega} H(\phi) dx. \end{aligned} \quad (3.26)$$

Again, from here we consider the case $\nu = 0, \lambda_1 = \lambda_2 = \lambda$. To compute the Euler-Lagrange equations of (3.26) the authors [33] introduce regularised version of H and

δ , to deal with the fact that the above is not differentiable at $\phi = 0$. The respective choices are given as

To compute the Euler-Lagrange equation for the unknown function ϕ , as H is not differentiable at 0, we consider regularised versions of H and δ functions, denoted by H_ϵ and δ_ϵ respectively [33]:

$$H_\epsilon(x) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan\left(\frac{x}{\epsilon}\right) \right), \quad \delta_\epsilon(x) = H'_\epsilon(x) = \frac{\epsilon}{\pi(\epsilon^2 + x^2)}, \quad (3.27)$$

where $H_\epsilon \rightarrow H$ when $\epsilon \rightarrow 0$. These are shown in Fig. 3.3. Then the regularised functional is given by

$$F_\epsilon^{LS} = \lambda \int_{\Omega} (z(x) - c_1)^2 H_\epsilon(\phi) dx + \lambda \int_{\Omega} (z(x) - c_2)^2 (1 - H_\epsilon(\phi)) dx + \mu \int_{\Omega} \delta_\epsilon(\phi) |\nabla \phi| dx. \quad (3.28)$$

The new minimisation problem is then given by

$$\min_{\phi, c_1, c_2} F_\epsilon^{LS}(\phi, c_1, c_2). \quad (3.29)$$

The minimisers with respect to the intensity constants can be given explicitly (with ϕ fixed) as follows:

$$c_1(\phi(x)) = \frac{\int_{\Omega} z(x) H_\epsilon(\phi(x)) dx}{\int_{\Omega} H_\epsilon(\phi(x)) dx}, \quad c_2(\phi(x)) = \frac{\int_{\Omega} z(x) (1 - H_\epsilon(\phi(x))) dx}{\int_{\Omega} (1 - H_\epsilon(\phi(x))) dx}. \quad (3.30)$$

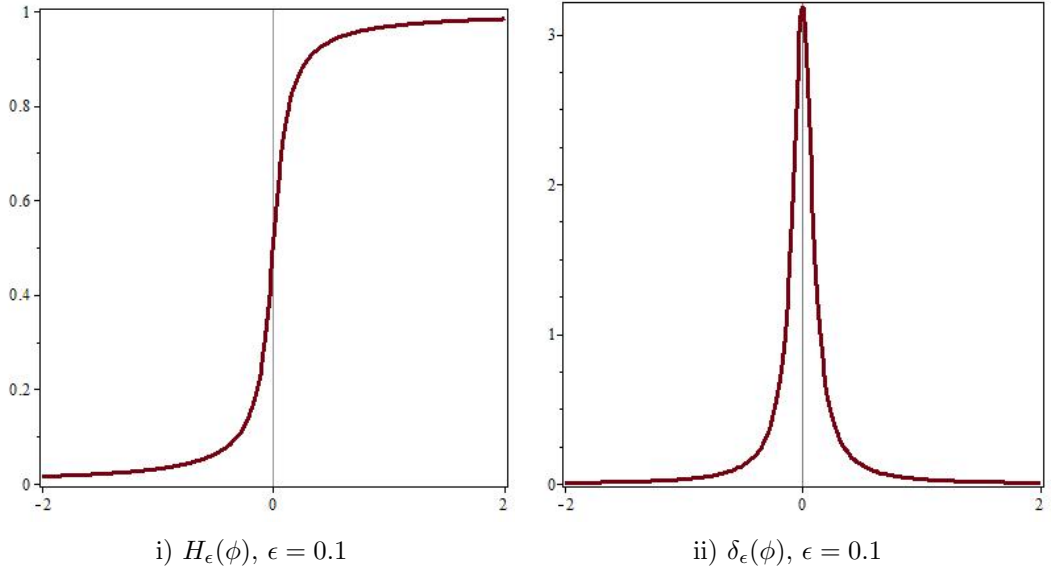


Figure 3.3: Approximation to the Heaviside and Delta functions with H_ϵ and δ_ϵ .

The authors derive the following Euler-Lagrange equation for ϕ (keeping c_1 and c_2 fixed):

$$\begin{cases} \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (z - c_1)^2 + \lambda_2 (z - c_2)^2 \right] = 0 & \text{in } \Omega, \\ \frac{\partial \phi}{\partial n} = 0 & \text{on } \partial \Omega. \end{cases} \quad (3.31)$$

Details of deriving the Euler-Lagrange equation are given in Chapter 2, or in the original paper [33]. The authors use a gradient descent scheme to solve the equation above. A multiphase formulation of the Chan-Vese model can be found in [123] and an extension to the piecewise-linear and piecewise-smooth formulations is given by [34].

The Chan-Vese functional is nonconvex with respect to ϕ , such that even if the intensity constants are known a priori the model can sometimes fail to successfully segment the image. This is dependent on the initialisation of the contour, Γ (or $\phi = 0$). We address methods that overcame this drawback next, introducing convex relaxation methods.

3.2.4 Convex Relaxation Methods

Despite the success of the Chan-Vese model, avoiding the presence of local minima was still a challenge. This problem is inherent to piecewise-constant segmentation in this framework as it involves minimising functionals over characteristic functions of sets, even when the intensity constants are known. In 2006 Chan, Esedoglu, and Nikolova [30] introduced an algorithm to find the global minimum of the two-phase piecewise-constant segmentation problem with fixed intensity constants. In this section, we will briefly discuss the idea behind this work and present some of the important details related to it.

First, we recall the Chan-Vese functional:

$$CV(\phi) = \int_{\Omega} |\nabla H_\epsilon(\phi)| dx + \lambda \int_{\Omega} (z(x) - c_1)^2 H(\phi) + (z(x) - c_2)^2 (1 - H(\phi)) dx.$$

Minimising the functional with respect to ϕ and applying a gradient descent scheme leads to

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

where $r(x) = (z(x) - c_1)^2 - (z(x) - c_2)^2$. The stationary solution of the above is the same as for

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

because $H_\epsilon(\phi)$ is a noncompactly supported, smooth approximation of the Heaviside function. Crucially, this gradient descent scheme corresponds to the following energy:

$$\int_{\Omega} |\nabla \phi| dx + \lambda \int_{\Omega} r(x) \phi dx.$$

In general, this energy does not have a minimiser as it is homogeneous of degree 1 in ϕ . By restricting the minimisation such that $0 \leq \phi(x) \leq 1$ for all $x \in \Omega$, and following the

work of Strang [119] Chan, Esedoglu, and Nikolova introduced the following theorem:

Theorem 3.2.1 *For any given fixed $c_1, c_2 \in \mathbb{R}$, a global minimiser for $MS(\cdot, c_1, c_2)$ can be found by carrying out the following convex minimisation*

$$\min_{0 \leq u(x) \leq 1} \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} r(x)u(x) dx \right\}$$

and then setting $\Sigma = \{x : u(x) \geq \mu\}$ for a.e. $\mu \in [0, 1]$.

A proof is provided in [30]. In Bresson et al. [18], the authors show that a global minimiser can be found when considering a weighted TV regulariser.

It is important to clarify here that a global minimiser can only be found when c_1 and c_2 are fixed. A completely convex formulation has been addressed by Brown et al. [20], which amounts to a convex relaxation of the K-means algorithm. However, in practice, if an image can be considered piecewise-constant then sufficiently accurate approximations of the intensity constants is trivial. Typically when the fitting function contains parameters that have to be optimised the joint problem is nonconvex, which can cause difficulties and is an area requiring further investigation. In this thesis we consider both cases, i.e. known and unknown fitting functions. In the next chapter we discuss new approaches for computing a global minimiser in the two-phase case. Here, we assume that the fitting function is known.

Also assumed with the ideas introduced above is that there is a foreground and background (i.e. two-phase). In this thesis, we consider segmentation problems of this type, but convex relaxation methods are applicable in a wider context and are very popular for multi-phase segmentation problems. Many approaches have been developed based on these methods [19, 78, 79, 25] or the analogous continuous max-flow approach [10, 136]. However, this work tends to require one segmentation function per region, which can be prohibitive for a large number of regions. Additionally, they assume that the number of regions is known, which isn't necessarily the case in many applications. Zhang et al. [141] proposed a novel approach based on the four-colour theorem [141], where the number of phases was globally optimised with only four segmentation functions.

Globally convex segmentation has proven to be a very effective technique in the past ten years. To summarise, in the context of two-phase segmentation where the fitting function (which we will call $f(x)$ from here) is known, we define the problem as follows:

$$\min_u \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx \right\}, \quad (3.32)$$

where $u \in BV(\Omega; [0, 1])$ and $f(x)$ is assumed to be some measurable function that takes positive and negative values. In the next chapter we will discuss a new approach to computing the global minimiser of this problem, which is applicable to the above and a wide range of closely related problems which will be addressed later in the thesis. In the following we will discuss some examples of approaches to find the compute the minimiser to (3.32).

3.3 Algorithms with Applications to Imaging

An important consideration in image processing problems is determining the most efficient way to compute a solution. In the level set formulation the contour is initialised as a distance function, that either needs to be periodically reinitialised [33] or have additional constraints applied [83] in order to maintain numerical stability. This is often time consuming. Furthermore, as discussed earlier in the chapter they are formulated in a nonconvex way, such that the global minimiser of the problem may not be found depending on initialisation. The introduction of the convex relaxation framework has overcome this problem in the sense that a gradient descent scheme is used in [30] to compute the global minimiser of the two-phase piecewise-constant Mumford-Shah formulation. However, gradient descent can be limited in the sense that the stability restrictions on the time step dictate that there is slow convergence. In recent years alternative formulations have been developed that have reduced the computation time for the main problem we consider (3.32). In this section we provide an overview of related approaches, and detail one method in particular. That is, the dual formulation of Chambolle [23] originally applied in this setting by Bresson et al. [18]. In Chapter 4 we propose a new method to minimise (3.32), based on a variation of an additive operator splitting [128, 85] scheme, that performs comparably well compared to [23, 18].

An important method to consider is that of Split Bregman, first introduced by Goldstein and Chan [59] for a general class of L1 regularised problems. In 2010, Goldstein, Bresson, and Osher [58] applied this method to a number of important models in image processing including ROF denoising [109] and the Geodesic Active Contours model [22]. Notably to our work they also applied it to globally convex segmentation (3.32). Crucially, their method avoids the regularisation term by introducing an auxiliary variable \vec{d} and a Bregman iteration, given by \vec{b} , to strictly enforce the constraint $\vec{d} = u$ giving the following sequence of optimisation problems:

$$(u^{k+1}, \vec{d}^{k+1}) = \min_{0 \leq u \leq 1, \vec{d}} \left\{ |\vec{d}| + \lambda \int_{\Omega} f u \, dx + \frac{\mu}{2} \|\vec{d} - \nabla u\|^2 \right\} \quad (3.33)$$

$$\vec{b}^{k+1} = \vec{b}^k + \nabla u^k - \vec{d}^k. \quad (3.34)$$

Their alternate minimisation scheme consists of computing an approximate solution of (3.33) with respect to u . Further details on the theoretical justification for an approximate solution at this stage can be found in [58] and [134]. The solution of (3.33) with respect to \vec{d} is given by a formula based on the shrink operator. The bregman update is given explicitly by (3.34) as with similar methods [126, 142]. A full description of the implementation of this approach and its algorithm can be found in [59, 58]. Impressive results are observed in comparison to [33] and [18], both in terms of computation time and dependence on the thresholding procedure discussed in the previous section.

Another important algorithm applicable to our main problem (3.32) is that of Chambolle-Pock [26], first introduced in 2011. It is a first order primal-dual algorithm

that is applicable to a wide range of problems in imaging. In [26], it is applied to ROF denoising [109], and related models in image deconvolution, inpainting, and motion estimation. Again, with our main problem (3.32) in mind we note that Chambolle-Pock is also applicable to globally convex segmentation. Furthermore, it is also suitable for multi-phase segmentation which offers a significant advantage. The problem (3.32) is reformulated in a saddle-point structure with the introduction of a new variable, analogous in some sense to the Split Bregman method discussed above. Details of the reformulation can be found in [26] and subsequent work [25], but we will not go into more depth here.

In the following section we summarise the details contained in [23] and [18], to introduce Chambolle's dual formulation of (3.32). Unlike Split-Bregman [58] and Chambolle-Pock [26], the dual formulation [23, 18] retains a regularisation approach similar to the original problem [30] whilst employing a splitting scheme. This approach is more relevant to our work, which we address next.

3.3.1 Chambolle's Dual Formulation

The dual formulation method of Chambolle [23] consists of introducing a new variable v and alternating between minimising u and v . It was first applied to segmentation problems of this type by Bresson et al. [18]. By splitting the variables in this way, the minimisation of u concentrates on the TV term, and the minimisation of v satisfies the fitting and constraint requirements:

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

where $\psi(v) = \max\{0, 2|v - \frac{1}{2}| - 1\}$. Two parameters are introduced here: $\theta > 0$ is a small parameter, and $\alpha > \frac{\lambda}{2} \|r(x)\|_{L^\infty(\Omega)}$ ensures the constraints on the indicator function $u(x)$ in (8.19) are met. The minimisation of u and v can be achieved iteratively by the following steps. With fixed v :

$$\min_u \left\{ \int_{\Omega} |\nabla u(x)| \, dx + \frac{1}{2\theta} \int_{\Omega} (u(x) - v(x))^2 \, dx \right\}$$

which can be solved by [23]:

$$u(x) = v(x) - \theta \nabla \cdot \rho(x), \tag{3.35}$$

where $\rho = (\rho^1, \rho^2)$ is the solution of

$$\nabla(\theta \nabla \cdot \rho - v) - |\nabla(\theta \nabla \cdot \rho - v)|\rho = 0, \tag{3.36}$$

which can be solved by a fixed point method $\rho^0 = 0$ and

$$\rho^{n+1} = \frac{\rho^n + \tau \nabla(\nabla \cdot \rho^n - v/\theta)}{1 + \tau |\nabla \rho^n - v/\theta|}. \tag{3.37}$$

With fixed u :

$$\min_v \left\{ \frac{1}{2\theta} \int_{\Omega} (u(x) - v(x))^2 dx + \int_{\Omega} \lambda r(x)v(x) + \alpha\psi(v) dx \right\},$$

given, based on the work of Bresson et al. [18], by:

$$v(x) = \min\{\max\{u(x) - \theta\lambda r(x), 0\}, 1\}. \quad (3.38)$$

We use this approach in Chapter 4 for comparative purposes, and in Chapter 6 following the method of D. Chen et al. [37].

Chapter 4

Additive Operator Splitting for Globally Convex Segmentation

4.1 Introduction

In this chapter we introduce an additive operator splitting (AOS) scheme for a two-phase segmentation problem in the convex relaxation framework. This builds on the content of the previous chapter where the idea of convex relaxation methods was introduced, and discussed in terms of its original application to imaging problems of this type in the form of Chan-Vese [33]. However, since the work of Chan, Esedoglu, and Nikolova [30] convex relaxation methods have been applied to a wide range of segmentation problems; this includes shape priors [101], intensity inhomogeneity [37], multi-phase [25, 10, 137, 102], and many more [92, 141, 75, 20]. Here we focus on a two-phase formulation with a generalised fitting function. Two-phase problems have the advantage of being widely applicable in a number of important fields, such as organ selection in medical images, and the consideration of problems of this type will form the basis of the rest of the work in this thesis.

Our work here consists of introducing a new penalty function to impose the relaxed constraint, $u \in [0, 1]$. This is important as the final segmentation contour, Γ , is given by a thresholding procedure that relies on the computed segmentation function being approximately binary. If this relaxed binary constraint is imposed in an unreliable way, then Γ is more parameter dependent. The intention is to have the implementation of convex relaxation methods be consistent with the theory [30], especially with respect to this binary consideration.

The main idea in this chapter is based on the introduction of an improved AOS scheme for problems of this type. That is, in the case of two-phase segmentation problems with a generalised fitting function. AOS has been used for similar segmentation problems [105, 104, 9], but this was in the level-set framework introduced in Chapter 2. Here, we apply it to globally convex segmentation (GCS) for the first time. The main challenge consists of incorporating the penalty function in such a way that we can achieve a stable convergence reliably. Additionally, reducing computation time with this consideration is of particular interest.

In Section 4.2 we introduce the idea of GCS, and discuss the general problem we

will consider in this chapter. We will then cover two methods to compute a solution of this problem, in order to put our approach in context. The first is gradient descent which was used in the original paper by Chan, Esedoglu, and Nikolova [30]. The second is the dual formulation of Chambolle [23] which was used in the work of Bresson et al. [18] and introduced in the previous chapter. We choose these as they both impose the relaxed binary constraint with a penalty function, similarly to our approach, and the dual formulation [23] is a very popular method, widely used in the literature [18, 37], and is considered to be an effective scheme. In Section 4.3 we provide some relevant details of finding global minimisers for the general GCS problem. Here, we introduce our proposed penalty function to impose the relaxed binary constraint. We demonstrate the convexity of this term for certain values of the regularisation parameter, such that the functional is convex in practice. We also give some details of the derivation of the corresponding Euler-Lagrange equation, before introducing our proposed AOS scheme in Section 4.4. We propose two methods; the first is focused on reducing computation time, and the second is based on ensuring a stable convergence. In Section 4.5, we compare our proposed methods from Sections 4.3 and 4.4 to the Chambolle’s dual formulation for GCS [18], and discuss its relation to gradient descent in [30]. The main considerations in our tests will be their accuracy in terms of segmentation, how close to binary the solutions are, and the computation time. We make some concluding remarks in Section 4.6. The methods introduced in this chapter are applicable to a wide range of problems and will be referred to throughout the thesis.

4.2 Globally Convex Segmentation

In this chapter we discuss the problem of finding global minimisers of segmentation models in general. This area was first discussed in the context of two-phase piecewise-constant segmentation by Chan, Esedoglu, and Nikolova [30] in 2006, which concerned finding the global minimum of the Chan-Vese functional [33] for fixed intensity constants. As discussed previously, this is a particular case of the Mumford-Shah functional [89] and further details can be found in Chapter 3. This seminal work [30] has influenced many important developments in this area, such as Bresson et al. [18] in 2007, as well as many other [79, 10, 136, 19, 120]. We are concerned with the two-phase segmentation problem, which in general is given as follows:

$$\min_{u \in \{0,1\}} \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx \right\}, \quad (4.1)$$

where $u(x) \in BV(\Omega; \{0, 1\})$ is a binary function that indicates the foreground and background for $u = 1$ and $u = 0$, respectively. The fitting term $f(x) \in L^{\infty}(\Omega)$ defines the boundary of the object in the sense that the zero level-set of this function is approximately the desired boundary Γ . Generally, $f(x) < 0$ indicates foreground and $f(x) > 0$ indicates background, with this balanced by the regularisation term and the weighting parameter, λ . The total variation (TV) of a binary function gives the length of the boundary, Γ , such that $TV(u) \geq 0$. We can then observe that the functional attains its

minimum when positive values of $f(x)$ are limited due to the indicator function going to $u = 0$. The weighting parameter, λ , then determines the smoothness of Γ . In this chapter we do not discuss how this fitting function, $f(x)$, is found. Many methods deal with what the best choice is in a number of problems, such as Mylona et al. [90].

In order to efficiently and reliably obtain the global minimum we need to minimise a convex functional over a convex set, which we discuss in Chapter 2. The above functional is convex with respect to u . However, the constraint set $\{0, 1\}$ is clearly nonconvex. The central idea here is that of convex relaxation, where the constraint is relaxed such that the set is convex. The conventional relaxation here is to allow the indicator function to take intermediate values, i.e. $u(x) \in [0, 1]$. We are now minimising a convex functional over a convex set, and so can obtain a global minimum. The advantage of formulating a segmentation model in this way is that success is determined by the fitting function and parameter selection, as there is no risk of obtaining a local minimum. This gives us the convex relaxation segmentation formulation:

$$\min_{u \in [0,1]} \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx \right\}. \quad (4.2)$$

This minimisation problem is central to the work discussed in this chapter. It has many possible applications, and understanding the background and seminal ideas surrounding it are essential to variational segmentation models. In what follows we discuss conventional methods to minimise this functional (4.2). Originally, Chan, Esedoglu, Nikolova [30] used gradient descent to obtain a solution. Many recent methods have approached this problem differently, as discussed in Section 3.3. Alternative minimisation methods for this type of problem include Goldstein et al. [58], and Chambolle-Pock [26].

Once a solution for the segmentation function u has been obtained, which we refer to throughout the thesis as u^* , the computed contour can be defined by the thresholding procedure described in [30] and given as follows. The foreground, Ω_1 is defined as

$$\Omega_1 = \Sigma(\gamma) = \{x : u(x) > \gamma\}. \quad (4.3)$$

The final segmentation contour can then be defined as $\Gamma^* = \partial\Omega_1$, and is a minimiser for the corresponding nonconvex problem for $\gamma \in (0, 1)$ [30].

4.2.1 Gradient Descent

In Chan, Esedoglu, and Nikolova [30] they impose the constraint in the functional (4.2) by including a penalty function, $\nu(u) = \max\{0, 2|u - 1/2| - 1\}$. Then, the unconstrained minimisation with respect to u is

$$\min_u \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx + \alpha \int_{\Omega} \nu(u) dx \right\}. \quad (4.4)$$

Further details can be found in [29, 30], or by referring to the discussion in Chapter 3. In order to smooth the kinks at $u = 0$ and $u = 1$ they introduce a regularised penalty function, such that the Euler-Lagrange equation can be derived. The regularised penalty function, $\psi_{\epsilon}(u)$ is defined as:

$$\psi_\epsilon(\zeta) = \begin{cases} -\zeta & \text{if } \zeta < -\epsilon/\sqrt{2} \\ (1 + \sqrt{2})\zeta - \sqrt{\tan^2(3\pi/8)\zeta^2 - (\zeta - \epsilon)^2} & \text{if } -\epsilon/\sqrt{2} \leq \zeta < \epsilon \\ 0 & \text{if } \epsilon \leq \zeta < 1 - \epsilon \\ (1 + \sqrt{2})\zeta - \sqrt{\tan^2(3\pi/8)\zeta^2 - (\zeta - 1 + \epsilon)^2} & \text{if } 1 - \epsilon \leq \zeta < 1 + \epsilon/\sqrt{2} \\ \zeta - 1 & \text{if } 1 + \epsilon/\sqrt{2} \leq \zeta \end{cases}$$

This was first introduced by Bresson [17] in 2005. This gives the new minimisation problem:

$$\min_u \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx + \alpha \int_{\Omega} \psi_\epsilon(u) dx \right\}. \quad (4.5)$$

The following Euler-Lagrange equation with Neumann boundary conditions is derived:

$$\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \lambda f - \alpha \psi'_\epsilon(u) = 0, \quad \frac{\partial u}{\partial \vec{n}} = 0. \quad (4.6)$$

The details of this derivation will be discussed in Section 4.3.3, where we refer to the mathematical preliminaries introduced in Chapter 2. This nonlinear partial differential equation (PDE) (4.6) can be solved with an explicit gradient descent method by introducing a time step:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \lambda f - \alpha \psi'_\epsilon(u), \quad \frac{\partial u}{\partial \vec{n}} = 0. \quad (4.7)$$

In [29, 30] they present results for this approach using gradient descent. As discussed in Chapter 2 the restrictions on the time step can be limiting in terms of computation time. For example, Bresson [17] uses a time step of $\tau = 0.00005$. The converged results for $u(x)$ are close to binary for the results presented in [30], where the authors state that "the extreme values seem to be about 0.04 at the low end and 0.97 at the high end". We will consider these details in the introduction of our proposed methods, in Sections 4.3 and 4.4.

4.2.2 Dual Formulation

The dual formulation of Chambolle [23] was first applied to this problem type by Bresson et al. [18] in 2007. We discuss it in Chapter 3 but recall the essential details here in order to compare against our proposed method in Section 4.4. The Chambolle Algorithm consists of introducing a new variable $v(x)$ and alternating between minimising u and v . By splitting in this way the minimisation of u concentrates on the TV term, and the minimisation of v satisfies the fitting and constraint requirements. The original functional (4.2) is adjusted as follows:

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

where $\theta > 0$ is a small weighting parameter. The minimisation of u and v is achieved by alternating between minimising each and iterating. First, with a fixed v , the functional is minimised with respect to u :

$$\min_u \left\{ \int_{\Omega} |\nabla u(x)| dx + \frac{1}{2\theta} \int_{\Omega} (u(x) - v(x))^2 dx \right\}.$$

This can be solved for u according to [23], as detailed in the previous chapter. Secondly, with fixed u , the functional is minimised with respect to v :

$$\min_v \left\{ \frac{1}{2\theta} \int_{\Omega} (u(x) - v(x))^2 dx + \int_{\Omega} \lambda f(x)v(x) + \alpha v(x) dx \right\}.$$

Based on the work of Bresson et al. [18], this can be solved by:

$$v(x) = \min\{\max\{u(x) - \theta\lambda f(x), 0\}, 1\}.$$

Minimising with respect to u and v is iterated until convergence. The stopping criterion given in [18] is $\max(|u_{n+1} - u_n|, |v_{n+1} - v_n|) \leq \delta$, although a suitable value of δ is not given. We will address this in our test sets in Section 4.5. Chambolle's dual formulation is widely used in the literature [37, 18] in relation to the original problem 4.2, and is recognised as an effective approach both in terms of accuracy and computational efficiency.

4.3 Finding the Global Minimum

Our approach to the two-phase GCS problem (4.2) is to follow the seminal work of [30], where the relaxed binary constraint $u \in [0, 1]$ is enforced with a penalty function. The corresponding unconstrained minimisation problem is given as follows:

$$\min_{u \in [0,1]} \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx + \alpha \int_{\Omega} \nu(u) dx \right\}, \quad (4.8)$$

where $f(x)$ is a fitting term weighted by $\lambda > 0$, and $\nu(u)$ is an exact penalty term weighted by $\alpha > 0$. In [17, 29, 30] this function, shown in Fig. 4.1, is given as

$$\nu(u) = \max\{0, 2|u - 1/2| - 1\}. \quad (4.9)$$

The choice of α is important to the imposing the constraint $0 \leq u \leq 1$ in a robust way. In the following we explore an alternative regularisation of the penalty function $\nu(u)$, which we intend to improve results in the sense of how binary the converged result is and how fast it is obtained.

4.3.1 Introducing a New Regularised Penalty Function

Whilst many recent methods have enforced the relaxation constraint on u in alternative ways [25, 75, 26, 58], we return to the original approach where a penalty function is used. As discussed in Chapter 3, $\nu(u)$ is an exact penalty term and enforces the constraint provided that $\alpha > \frac{\lambda}{2} \|f(x)\|_{L^\infty}$. The constrained and unconstrained energies,

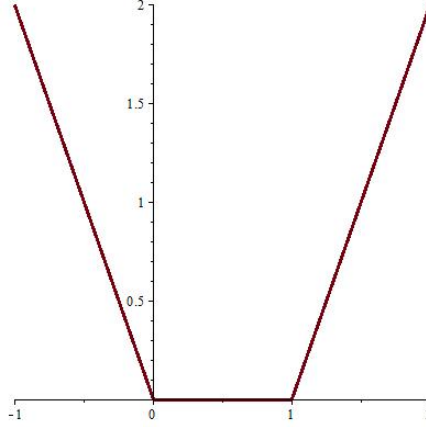


Figure 4.1: The penalty function $\nu(u)$ used in [17, 29, 30] to enforce the constraint $u \in BV(\Omega; [0, 1])$.

given in (4.2) and (4.8) respectively, agree for $\{u \in L^\infty(\Omega) : 0 \leq u(x) \leq 1 \ \forall x\}$. If $\alpha > \frac{\lambda}{2} \|f(x)\|_{L^\infty}$, then

$$|\lambda f(x)| \max\{|u(x)|, |u(x) - 1|\} < \alpha \nu(u(x)), \quad \text{whenever } u(x) \in [0, 1],$$

meaning the transformation $u \rightarrow \min\{\max\{0, u\}, 1\}$ always decreases the energy of the unconstrained problem. Therefore, any minimiser of the unconstrained problem automatically satisfies the constraint $0 \leq u \leq 1$. In [30] a regularised penalty function is used, and they discuss the results using gradient descent regarding how close to binary the solution for u is. They find that as the steady state approaches u becomes approximately binary, despite taking a continuum of values during the evolution. The "extreme values seem to be about 0.04 at the low end and 0.97 at the high end", which is as a result of the regularisation of the problem. The thresholding procedure works in practice in this case, but clearly is not consistent with $\Sigma(\gamma) = \{x : u(x) > \gamma\}$ for $\gamma \in (0, 1)$. Ideally, the regularisation of the functional should be as consistent with the theoretical problem as possible and we now consider possible improvements to the formulation in this sense. With this in mind, we introduce a new penalty function that addresses this problem.

We introduce a new function, $\nu_\epsilon(u)$ that is based on an intuitive definition of the penalty function based on its shape in Fig. 4.1. It is given as follows:

$$\nu_\epsilon(u) = H_\epsilon\left(\sqrt{(2u-1)^2 + \epsilon} - 1\right) \left[\sqrt{(2u-1)^2 + \epsilon} - 1\right],$$

where $H_\epsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi} \arctan \frac{x}{\epsilon}\right)$. Examples for given choices of ϵ are shown in Fig. 4.2. This gives us a new unconstrained minimisation problem:

$$\min_{u(x)} \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx + \alpha \int_{\Omega} \nu_\epsilon(u) dx \right\}. \quad (4.10)$$

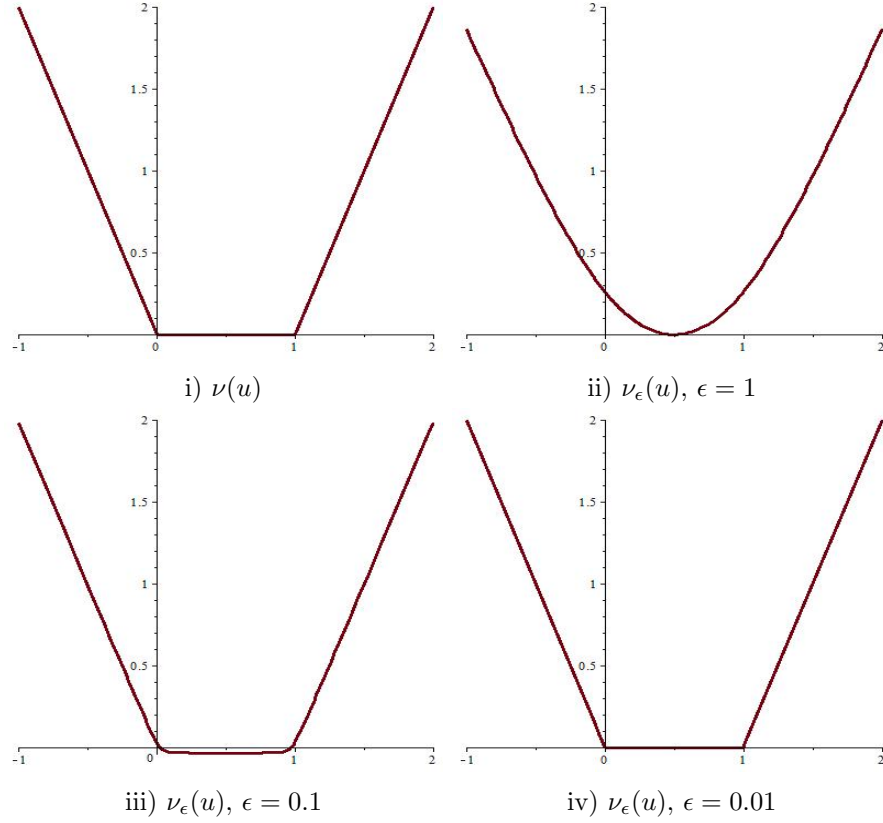


Figure 4.2: The regularised penalty function $\nu_\epsilon(u)$ for ii) $\epsilon = 1$, iii) $\epsilon = 0.1$, and iv) $\epsilon = 0.01$. The original penalty function, $\nu(u)$, from [30] is shown in i). Visually, the most appropriate choice is for $\epsilon = 0.01$.

4.3.2 Convexity of the Proposed Functional

In introducing a new term, $\nu_\epsilon(u)$, to the original minimisation problem (4.2) it is important to ensure that it does not violate the most important condition of the functional. That is, the new functional must also be convex with respect to u otherwise finding the global minimum cannot be guaranteed with convex relaxation. In order to establish that this property is retained in (4.10) we briefly return to the mathematical preliminaries discussed in Chapter 2. Let us consider the proposed functional as the sum of two functions:

$$\mathcal{F}_\epsilon(u) = \mathcal{J}(u) + \mathcal{P}(u),$$

where

$$\mathcal{J}(u) = \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx, \quad \mathcal{P}(u) = \alpha \int_{\Omega} \nu_\epsilon(u) dx.$$

As the sum of two convex functions is also convex, we can demonstrate the convexity of $\mathcal{F}_\epsilon(u)$ by treating of $\mathcal{J}(u)$ and $\mathcal{P}(u)$ separately. As discussed in Chapter 2, we require the second order derivative to be non-negative in order to prove convexity. We will not show this for $\mathcal{J}(u)$, as it is already well established [29, 30, 17, 18]. It is, however, necessary to consider the second function, $\mathcal{P}(u)$, involving the proposed regularised

penalty function. First we introduce an alternative definition of $\nu_\epsilon(u)$:

$$\nu_\epsilon(u) = H_\epsilon(b_\epsilon(u))b_\epsilon(u),$$

where $b_\epsilon(u) = \sqrt{(2u-1)^2 + \epsilon} - 1$ and $H_\epsilon(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{\epsilon}\right)$. Then, the second derivative of $\mathcal{P}(u)$ is given as

$$\mathcal{P}_\epsilon''(u) = \alpha \frac{\partial^2 \mathcal{P}(u)}{\partial u^2} = \alpha \frac{\partial^2 H_\epsilon(b_\epsilon)}{\partial u^2} b_\epsilon(u) + 2\alpha \frac{\partial H_\epsilon(b_\epsilon)}{\partial u} \frac{\partial b_\epsilon(u)}{\partial u} + \alpha \frac{\partial^2 b_\epsilon(u)}{\partial u^2} H_\epsilon(b_\epsilon), \quad (4.11)$$

where

$$\begin{aligned} \frac{\partial b_\epsilon(u)}{\partial u} &= (4u-2)[(2u-1)^2 + \epsilon]^{-1/2}, \\ \frac{\partial^2 b_\epsilon(u)}{\partial u^2} &= 4[(2u-1)^2 + \epsilon]^{-1/2} - (4u-2)^2[(2u-1)^2 + \epsilon]^{-3/2}, \\ \frac{\partial H_\epsilon(b_\epsilon)}{\partial u} &= \frac{\epsilon}{\pi} \left(\frac{(4u-2)[(2u-1)^2 + \epsilon]^{-1/2}}{\epsilon^2 + ([(2u-1)^2 + \epsilon]^{1/2} - 1)^2} \right), \\ \frac{\partial^2 H_\epsilon(b_\epsilon)}{\partial u^2} &= \frac{\epsilon}{\pi} \left(\frac{4[(2u-1)^2 + \epsilon]^{-1/2} - (4u-2)^2[(2u-1)^2 + \epsilon]^{-3/2}}{\epsilon^2 + ([(2u-1)^2 + \epsilon]^{1/2} - 1)^2} \right) \\ &\quad - \frac{2\epsilon}{\pi} \left(\frac{(4u-2)[(2u-1)^2 + \epsilon]^{-1/2}}{\epsilon^2 + ([(2u-1)^2 + \epsilon]^{1/2} - 1)^2} \right)^2. \end{aligned}$$

From this we can see that determining whether (4.11) is non-negative is not trivial and currently we have not proved this for any $\epsilon > 0$ in general. What this function, $\mathcal{P}_\epsilon''(u)$, looks like is shown in Fig. 4.3 for $\epsilon = 1$ and $\epsilon = 0.1$, and we can observe the challenge in proving its non-negativity. Intuitively, $\mathcal{P}(u)$ is convex but we leave the problem of proving this for any $\epsilon > 0$ open for future consideration. However, for fixed values of ϵ we can observe that $\mathcal{P}_\epsilon''(u)$ is non-negative. This allows us to minimise (4.10) with the knowledge that the functional is convex, and the global minimum can be found accordingly.

4.3.3 Deriving the Euler-Lagrange Equation

We now discuss minimising (4.10) with respect to $u(x)$ to derive the Euler-Lagrange equation. From Chapter 2 we recall that $F_\epsilon(u)$ is differentiable in the Gateaux sense at $u \in BV(\Omega)$ if the limit

$$\mathcal{F}'_\epsilon(u; \phi) = \left. \frac{d}{dh} \left(\mathcal{F}_\epsilon(u + h\phi) \right) \right|_{h=0} = \lim_{h \rightarrow 0} \frac{\mathcal{F}_\epsilon(u + h\phi) - \mathcal{F}_\epsilon(u)}{h},$$

is defined for a test function ϕ . With that in mind we can compute the first variation of the functional F_ϵ with respect to u such that

$$\lim_{h \rightarrow 0} \frac{\mathcal{F}_\epsilon(u + h\phi) - \mathcal{F}_\epsilon(u)}{h}.$$

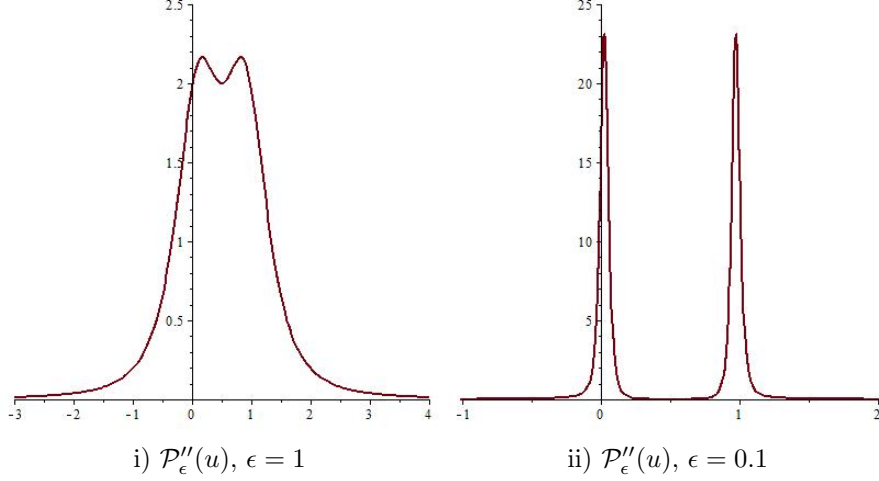


Figure 4.3: The second derivative of $\mathcal{P}(u)$, $\mathcal{P}''_\epsilon(u)$, given by (4.11). i) is for $\epsilon = 1$ and ii) is for $\epsilon = 0.1$. Both are non-negative and therefore the corresponding $\mathcal{F}_\epsilon(u)$ is a convex functional.

This is given as follows

$$\begin{aligned} \frac{d}{dh} \left(\int_{\Omega} |\nabla(u + h\phi)| dx + \int_{\Omega} f(x)(u + h\phi) dx + \int_{\Omega} \nu_\epsilon(u + h\phi) dx \right) \Big|_{h=0} &= 0 \\ \int_{\Omega} \frac{\nabla u}{|\nabla u|} \cdot \nabla \phi dx + \lambda \int_{\Omega} f \phi dx + \alpha \int_{\Omega} \nu'_\epsilon(u) \phi dx &= 0 \end{aligned}$$

where

$$\begin{aligned} \nu'_\epsilon(u) &= \frac{2\epsilon(2u-1)(\sqrt{(2u-1)^2 + \epsilon} - 1)}{\pi\sqrt{(2u-1)^2 + \epsilon}(\sqrt{(2u-1)^2 + \epsilon} - 1)^2 + 1} \\ &\quad + \frac{2H_\epsilon(\sqrt{(2u-1)^2 + \epsilon} - 1)(\sqrt{(2u-1)^2 + \epsilon} - 1)}{\sqrt{(2u-1)^2 + \epsilon}}, \end{aligned}$$

Recall from Gauss's Theorem in Chapter 2 the following relation:

$$\int_{\Omega} \phi \nabla \cdot \vec{w} dx = - \int_{\Omega} \nabla \phi \cdot \vec{w} dx + \int_{\partial\Omega} \phi \vec{w} \cdot \vec{n} ds.$$

We then have:

$$\begin{aligned} \int_{\Omega} \frac{\nabla u}{|\nabla u|} \cdot \nabla \phi dx + \lambda \int_{\Omega} f \phi dx + \alpha \int_{\Omega} \nu'_\epsilon(u) \phi dx &= 0 \\ \int_{\partial\Omega} \phi \frac{\nabla u}{|\nabla u|} \cdot \vec{n} ds - \int_{\Omega} \phi \nabla \cdot \frac{\nabla u}{|\nabla u|} dx + \int_{\Omega} (\lambda f + \alpha \nu'_\epsilon(u)) \phi dx &= 0 \\ \int_{\partial\Omega} \phi \frac{\nabla u}{|\nabla u|} \cdot \vec{n} ds + \int_{\Omega} \left(\lambda f + \alpha \nu'_\epsilon(u) - \nabla \cdot \frac{\nabla u}{|\nabla u|} \right) \phi dx &= 0. \end{aligned}$$

This holds for all test functions ϕ , giving us the following Euler-Lagrange equation with Neumann boundary conditions:

$$\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \lambda f - \alpha \nu'_\epsilon(u) = 0, \quad \frac{\partial u}{\partial \vec{n}} = 0. \quad (4.12)$$

This nonlinear PDE (4.13) can be solved with the gradient descent method by introducing a time step, as discussed in Section 4.2.1. However, in the next section we consider potential improvements to this method in the form of an AOS scheme.

4.4 A New Additive Operator Splitting Scheme for GCS

We consider the following PDE, based on the derivation from the previous section. We consider the 2D case, but it can be generalised to higher dimensions as discussed in Chapter 2.

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|_\beta} \right) - \lambda f - \alpha \nu'_\epsilon(u), \quad (4.13)$$

where $|\nabla u|_\beta = \sqrt{u_x^2 + u_y^2 + \beta}$, to avoid singularities [109]. Denote $W = \frac{1}{|\nabla u|_\beta}$. Freezing W linearises the equation and 4.13 can be rewritten in the form:

$$\frac{\partial u}{\partial t} = \partial_x(W \partial_x u) + \partial_y(W \partial_y u) - \lambda f - \alpha \nu'_\epsilon(u).$$

We apply the semi-implicit AOS scheme proposed by [85, 128], where the PDE (4.13) is linearised. Below, A_ℓ are the diffusion quantities in the ℓ direction ($\ell = 1, 2$ for each spatial dimension) and were derived using the finite difference method, and n denotes the n^{th} iteration. They are given explicitly in Chapter 5 and in our paper [112]. The benefits of this method are that at each iteration the solution to two tridiagonal linear systems is required, which can be computed efficiently with the Thomas algorithm [128, pp.5-6]. We will demonstrate the adjustments made for our new method by first looking at the important step of the original AOS scheme [128], that we call AOS0 from here:

$$u_\ell^{n+1} = \underbrace{(I - 2\tau A_\ell(u^n))^{-1}}_{Q_0} \underbrace{(u^n - \tau \alpha \nu'_\epsilon(u^n) - \tau \lambda f)}_{f_0}. \quad (4.14)$$

Then, the update is given by averaging the update from each direction:

$$u^{n+1} = \frac{u_1^{n+1} + u_2^{n+1}}{2}. \quad (4.15)$$

However, AOS0 generally assumes that f_0 is not dependent on u . In this case, the term $\nu'_\epsilon(u)$ in f_0 does depend on u , which can lead to stability problems in practice. This prompted us to consider an extension of the original scheme, to improve performance and ensure stability. The shape of $\nu'_\epsilon(u)$ means that changes are problematic near $u = 0$ and $u = 1$, as small changes in u produce large changes in f_0 .

4.4.1 Method 1

In order to overcome this, we define an interval I_ζ , where we adjust the equation based on the linear part of $\nu'_\epsilon(u)$ and the difference in u between iterations. This minimises

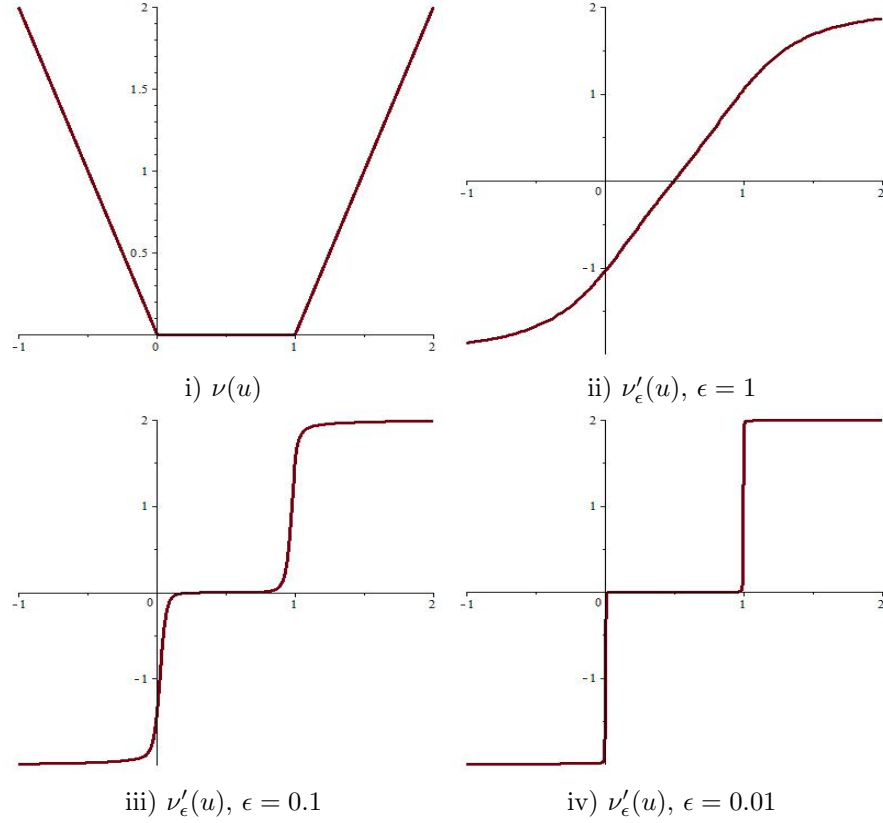


Figure 4.4: The function $\nu'_\epsilon(u)$ for different choices of ϵ . The jumps at $u = 0$ and $u = 1$ aren't as sharp for larger ϵ , but the constraint $u \in [0, 1]$ is enforced less strictly in these cases.

the changes in f_0 from n to $n + 1$. We make an adjustment to the equation based on the Taylor expansion of $\nu'_\epsilon(u)$ at $u = 0$; $\nu'_\epsilon(u) = a_0 + bu + \mathcal{O}(u^2)$, and at $u = 1$; $\nu'_\epsilon(u) = a_1 + bu + \mathcal{O}(u^2)$. This allows us to approximate $\nu'_\epsilon(u)$ in an interval, I_ς , with a linear function, bu . We define this interval and a binary function, \tilde{b}^n , as follows:

$$I_\varsigma := [0 - \varsigma, 0 + \varsigma] \cup [1 - \varsigma, 1 + \varsigma], \quad \tilde{b}^n = \begin{cases} b, & \text{if } u^n \in I_\varsigma \\ 0, & \text{elsewhere.} \end{cases} \quad (4.16)$$

Then, with $\tilde{B}^n = \text{diag}(\tau\alpha\tilde{b}^n)$, we can adjust (4.14):

$$u_\ell^{n+1} = \underbrace{\left(I + \tilde{B}^n - 2\tau A_\ell(u^n) \right)^{-1}}_{Q_1} \underbrace{\left(u^n + \tau\alpha\tilde{b}^n u^n - \tau\alpha\nu'_\epsilon(u^n) - \tau\lambda f \right)}_{f_1}. \quad (4.17)$$

This scheme improves the performance of AOS0 because the changes in f_1 (4.17) between iterations is limited, compared to f_0 (4.14). The addition of $\tau\alpha\tilde{b}^n u_1^n - \tau\alpha\tilde{b}^n u_1^{n+1}$ has the effect of approximating the change in $\nu'_\epsilon(u)$ between n and $n + 1$, in I_ς . We call the above scheme AOS1 (4.17) from here.

4.4.2 Method 2

In Weickert et al. [129] conditions on Q_ℓ for a discrete scale space were provided, required for convergence. The matrix Q_1 (4.17) does not fulfil this criteria and in order

to satisfy these conditions, we must first make the following adjustment, compared to (4.17):

$$u_\ell^{n+1} = \underbrace{\left(I - 2\tau(I + \tilde{B}^n)^{-1} A_\ell(u^n) \right)^{-1}}_{Q_2} \underbrace{\left(u^n - \tau(I + \tilde{B}^n)^{-1} (\alpha \nu'_\epsilon(u^n) + \lambda f) \right)}_{f_2}. \quad (4.18)$$

By increasing ς , such that $\tilde{b} = b$, AOS2 fulfils additional criteria from [129]. As $u \in [0, 1]$, $\varsigma = 0.5$ is enough to ensure this. This adjustment consists of multiplying τ by a scalar, dependent on b and α . This can be interpreted as automatically restricting the time step, based on the prominence of the penalty function, dictated by the size of α , and represented by b . Our results demonstrate a significant improvement in terms of speed for AOS1, and stability for AOS2, over the original AOS scheme applied to this type of problem (4.8).

The relevant conditions are listed in Chapter 2, and additional details can be found in our paper [112] or in the literature [127, 128, 129]. This adjustment consists of multiplying τ by a scalar, dependent on b and α . This can be interpreted as automatically setting the time step to $\tilde{\tau}$:

$$\tilde{\tau} = \frac{\tau}{1 + \tau \alpha b}. \quad (4.19)$$

This restricts the size of time step based on the prominence of the penalty function, dictated by the size of α , and represented by b .

4.5 Experimental Results

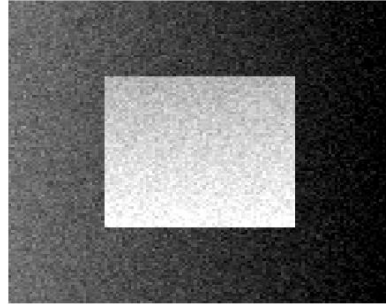
The test problems are given in Fig. 4.5 and show the image, z , the fitting function, f , and the zero contour of f (denoted Γ_f). One notes that the fitting function is given, which is not true in practice. We consider a generalised segmentation problem here, and in later chapters address how the fitting functions are formed for difficult problems. These tests are designed to compare our approach with Chambolle's Dual Formulation, which we call CDF from here. In the first set of tests we establish the parameters we use in our AOS schemes, and demonstrate the advantages of our proposed schemes. In the second test we compare AOS1 and AOS2 against CDF [23, 18]. In the third we demonstrate how dependent results are on the initialisation of the segmentation function, u_0 .

We use three measures to compare the two approaches. The accuracy of the segmentation, given by the Tanimoto Coefficient [48]:

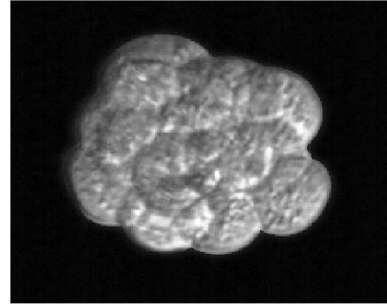
$$TC = \frac{N(GT \cap \Omega_1^*)}{N(GT \cup \Omega_1^*)}, \quad (4.20)$$

where $N(\cdot)$ is the number of pixels in the enclosed region, GT is the ground truth, and Ω_1^* is the result computed based on the thresholding procedure defined earlier and in [30]. We also want to consider how close to binary the computed solution is. With that in mind we define an interval, $I_\mu = [0, \mu] \cup [1 - \mu, 1]$ and then mb is the percentage of

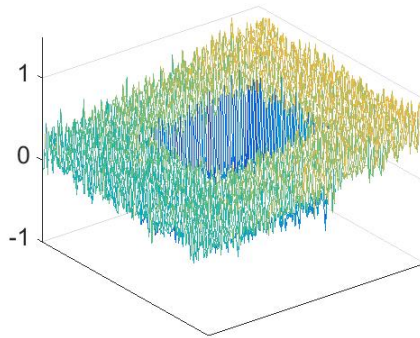
$x : u^*(x) \in I_\mu$. Typically we select $\mu = 0.02$, and also present histograms that use this spacing as well. We are also interested in the computation time, which we denote cpu (measured in seconds). Each approach has a stopping criterion, defined as δ , which will be discussed in Test Set 2.



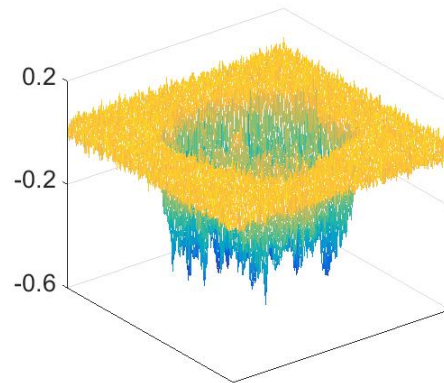
i) Image 1, $z(x)$



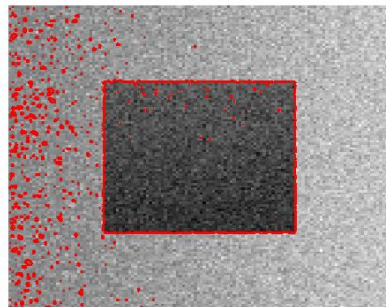
ii) Image 2, $z(x)$



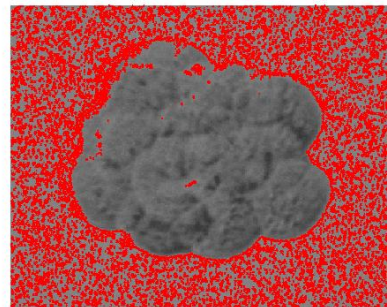
iii) Fitting 1, $f(x)$



iv) Fitting 2, $f(x)$



v) Fitting 1, Γ_f (red)



vi) Fitting 2, Γ_f (red)

Figure 4.5: Test Problems. Two examples are given for two-phase segmentation problems where the ground truth is known. Image 1 and 2 are on the left and right, respectively. Row 1 is the observed image, row 2 is the fitting function $f(x)$, and row 3 gives the zero contour, Γ_f , of $f(x)$ in red.

4.5.1 Test Set 1 (AOS Parameters)

In this set of tests we address the effectiveness of the ideas introduced in Sections 4.3 and 4.4, particularly with respect to parameter choices for the regularised penalty function, $\nu_\epsilon(u)$, and the AOS interval, I_ζ (4.16). We will present results for the original AOS scheme, and compare that against the proposed improved schemes: AOS1 and AOS2.

First, in Fig. 4.6 we show the computed segmentation function, $u^*(x)$, and its histogram (H) for three cases: $\epsilon = 1, 0.1, 0.01$. This will demonstrate how we select the parameter for the regularised penalty function introduced in Section 4.3.1. It is worth noting, as mentioned in Section 4.3.2, that for each choice of ϵ tested the second derivative of the functional is non-negative and so we expect to compute the global minimum for each example. Clearly the best result is for $\epsilon = 0.01$, particularly in terms of how binary the final result is. This is important as it means there is more consistency with the theory discussed in [30, 18, 17, 102] and the thresholding procedure to obtain the final contour is more reliable. This result is predictable based on the shape of the regularisations in Fig. 4.2. However, it is important to point out that smaller choices of ϵ are not as reliable, due to the discontinuity leading to numerical problems. The limits of this parameter are worth considering in future work. The key point from these tests is that we observe that for $\epsilon = 0.01$ the computed segmentation function is closer to binary than when the old penalty function is used in [30]. Chan et al. [30] state that "extreme values seem to be about 0.04 at the low end and 0.97 at the high end", whereas in our results the value of every pixel is in the interval I_μ , for $\mu = 0.02$.

In Fig. 4.7 we present results for AOS0, for three different time steps ($\tau = 10^{-3}, 10^{-2}, 10^{-1}$). Residuals (R) are presented along with the computed segmentation function, $u^*(x)$, for Image 1. We conclude that a small time step is required for the smooth convergence of $u(x)$ and for the result to be binary. Whilst this is a single example we observe this behaviour in both examples, and throughout other testing with this method.

In Table 4.1 we present results for our improved scheme, AOS1 ($\tau = 0.01$), based on selections of the interval parameter ζ . The stopping criterion used was $\delta = 0.01$. For Image 1 ($\zeta = 0, 0.001$) and Image 2 ($\zeta = 0, 0.001, 0.01$) the residual did not reduce to this level and the iterations were stopped at a maximum number. The results demonstrate that the choice of ζ is important for the speed of convergence and how close to binary the computed segmentation function $u^*(x)$ is. Clearly, we can use a higher time step than in AOS0 (see Fig. 4.7) and this contributes to the improvements achieved in the proposed scheme. The results in Fig. 4.8 support this conclusion. It illustrates what we've consistently observed in testing AOS1, where we present residuals for four choices of ζ with $\tau = 0.01$ (n.b. no stopping criterion was applied here). The optimal selection of this parameter warrants further investigation, as it is capable of quantifiably improving results.

In Fig. 4.9 we present results for the improved scheme, AOS2. We ran tests with a time step of $\tau = 1$, however these results have been replicated for higher values. We

present residuals for three selections of the interval parameter ($\varsigma = 0.01, 0.05, 0.5$) that demonstrate the advantages of this method. For $\varsigma = 0.5$ we consistently observe stable convergence and a computed segmentation function that is close to binary. Experimentally, the choice of τ is essentially arbitrary for AOS2 as mentioned in Section 4.4.2 (see eqn. (4.19)). However, the improvement is based on stability as increasing τ does not noticeably speed up convergence.

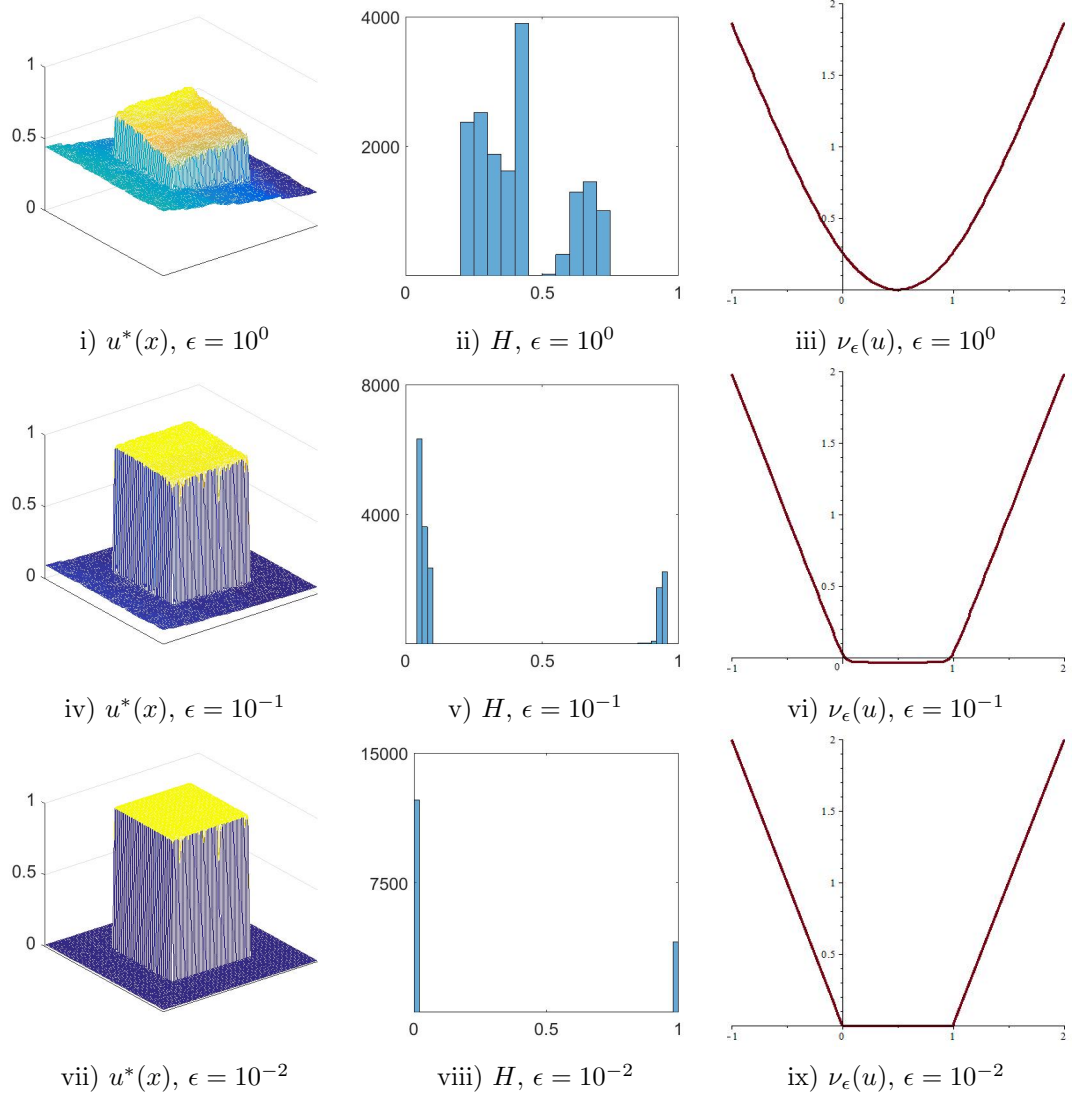


Figure 4.6: Test Set 1. ϵ Results, Image 1. The left column is the segmentation function, $u^*(x)$, the central column is the histogram of $u^*(x)$, and the right column is the regularised penalty function, $\nu_\epsilon(u)$. Row 1 is for $\epsilon = 10^0$, row 2 is for $\epsilon = 10^{-1}$, row 3 is for $\epsilon = 10^{-2}$. This demonstrates that a good choice for ϵ in the regularised penalty function, $\nu_\epsilon(u)$, is 10^{-2} . This is consistent throughout our tests, including for Image 2 which is not shown here.

4.5.2 Test Set 2 (Dual Formulation Comparison)

In this set of tests we compare the schemes proposed in Section 4.4, AOS1 and AOS2, against CDF for the original problem (4.2). We measure the success of each method

AOS1, Image 1				AOS1, Image 2			
ς	TC	mb	cpu	ς	TC	mb	cpu
0	1	69	318.3	0	0.96	58	1248.3
0.001	1	75	319.7	0.001	0.96	58	1254.4
0.01	1	100	1.4	0.01	0.96	95	1276.3
0.05	0.99	100	0.20	0.05	0.96	100	3.2
0.1	0.99	100	0.14	0.1	0.96	100	3.0
0.3	0.99	100	0.16	0.3	0.96	100	3.1
0.5	0.99	100	0.09	0.5	0.96	100	3.0

Table 4.1: Test Set 1. AOS1 ς Results for Images 1 (128x128) and 2 (256x256). In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). We present values as ς varies in terms of segmentation quality, TC , a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion $\delta = 0.01$, cpu (n.b. for some results the iterations were stopped at the maximum iteration number). Results demonstrate that smaller values of ς produce non-binary results and take longer to converge, despite the accuracy of the thresholding procedure.

with respect to the three features introduced above; that is, the accuracy (TC), the binary measurement (mb), and the computation time (cpu) in seconds. We note that a precise stopping criterion, δ , is not provided by the authors [18] and so we test two choices ($\delta = 0.1, 0.01$) with results presented in Tables 4.2 and 4.3. Empirically, we have found that a stopping criterion of $\delta = 0.01$ is sufficient for the proposed AOS schemes.

In Table 4.2 we present results for Image 1, which is of size 128x128, for a range of values of λ . This parameter controls the smoothness of the contour Γ and determines the accuracy of the model. Here, we compare AOS1 ($\varsigma = 0.05$) against CDF. One notes that a stopping criterion of $\delta = 0.1$ is sufficient for CDF here, as the results in terms of TC and mb do not improve for a lower threshold. We can observe that AOS1 is consistently more accurate than CDF. In fact, for $\lambda \in [5, 30]$ AOS1 achieves an impressive result of $TC = 0.99$, whereas CDF peaks at $TC = 0.97$. Similarly, AOS1 performs well in terms of mb for every value of λ . Conversely, the best result for CDF is $mb = 0.74$, meaning the thresholding procedure defined earlier is less reliable. Additionally, AOS1 performs better in terms of the computation time. AOS1 is less than a second for every value of λ , whilst CDF takes around ten seconds. It is also worth noting that the greatest cpu value for CDF ($\delta = 0.1$) corresponds with the best result in terms of accuracy. These conclusions are supported by Figs. 4.10 and 4.11

In Table 4.3 we present similar results for Image 2, which is of size 256x256. The conclusions from Table 4.2 are supported in these results, such as an appropriate stopping criterion for CDF being $\delta = 0.1$. The proposed scheme, AOS1 ($\varsigma = 0.05$), does not perform as well here as the first case. However, it still outperforms the alternative for $\lambda \in [5, 30]$ and is consistently good with respect to the TC and mb measures. This is supported, and demonstrated more clearly, by Figs. 4.12 and 4.13. We also note that the computation time of AOS1 is better than that of CDF, particularly for $\lambda = 5$ which is the best result in terms of accuracy for CDF.

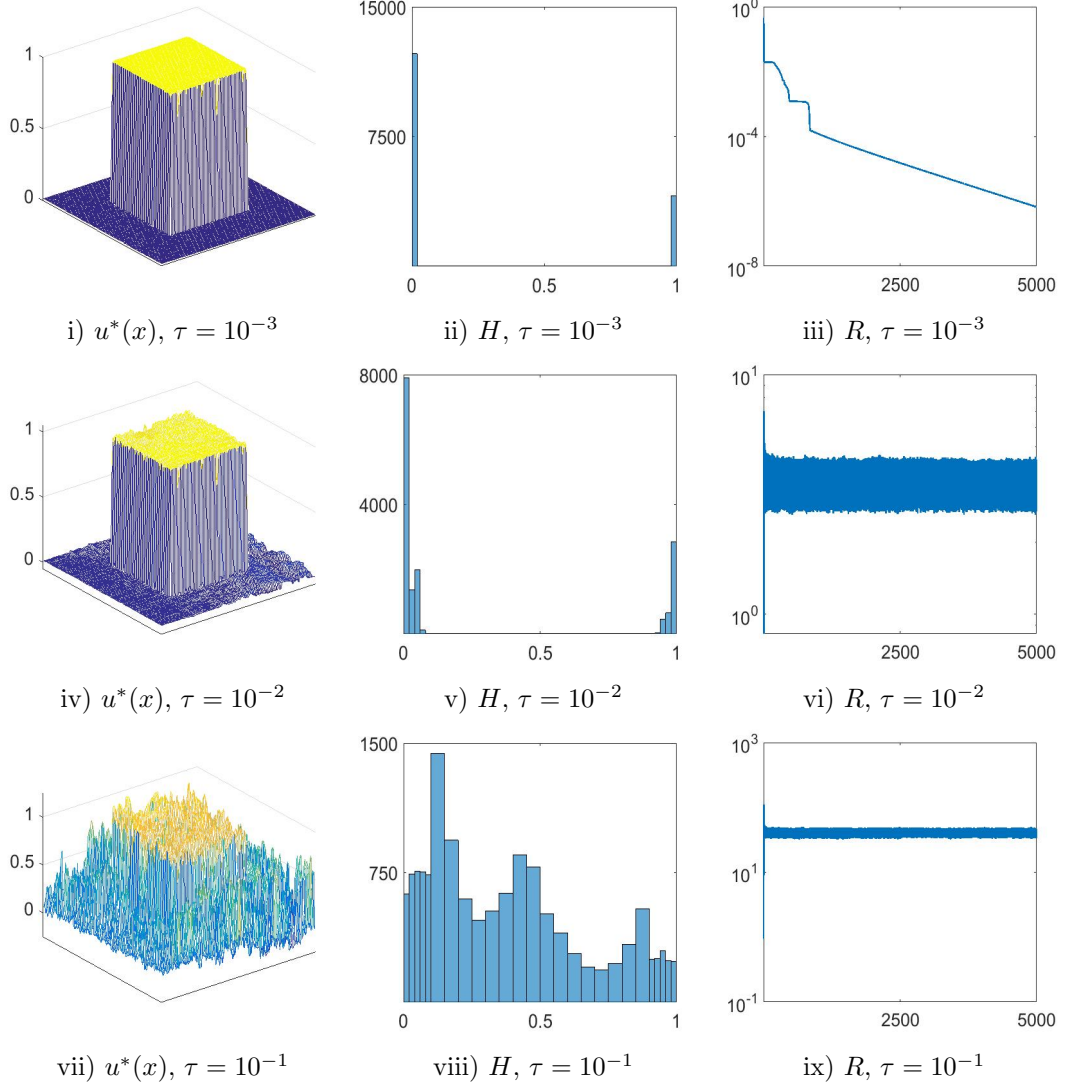


Figure 4.7: Test Set 1. AOS0 Results, Image 1. The left column is the segmentation function, $u^*(x)$, the central column is the histogram of $u^*(x)$, and the right column is the residual progression. Row 1 is for $\tau = 10^{-3}$, row 2 is for $\tau = 10^{-2}$, row 3 is for $\tau = 10^{-1}$. This demonstrates that for AOS0 a small time step ($\tau = 10^{-3}$) is required for a result that is close to binary, and a smooth convergence for $u(x)$. There are similar results for Image 2, which are not shown here.

4.5.3 Test Set 3 (Initialisation Dependence)

In this test set we address the significance of the initialisation of the segmentation function, u_0 . In this chapter we have established that global minimisers can be computed independently of initialisation, however, the choice of u_0 can be very important in terms of convergence speed. Here, we run the same tests as before for AOS1 with the stopping criterion $\delta = 0.01$

In Figs. 4.14 and 4.15 we introduce the initialisations for Images 1 and 2, respectively. The intention is to test a variety of choices, of varying distances from the ground truth, and observe the differences when varying ς . In Tables 4.4 and 4.5 we present results in terms of *mb* and *cpu*. One notes that the *TC* value of each, based on the

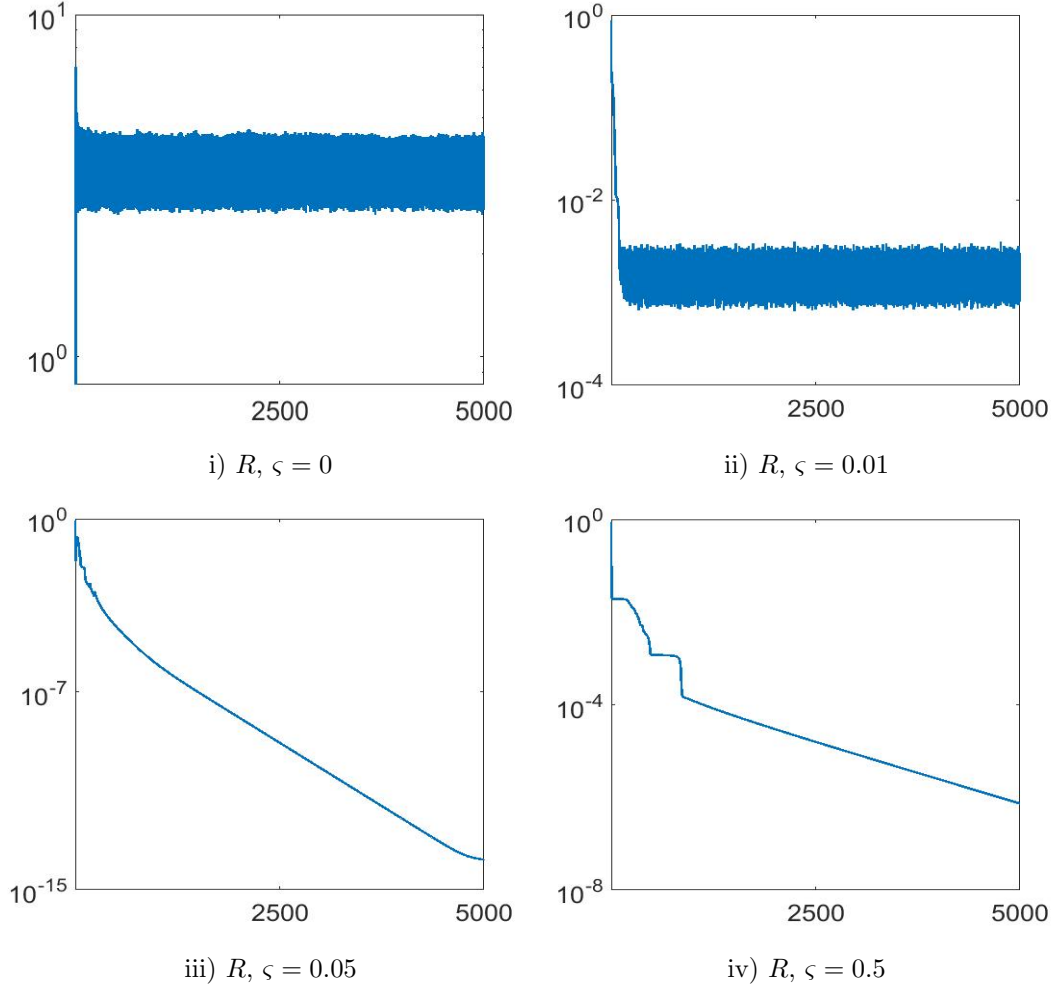


Figure 4.8: Test Set 1. AOS1 ς Results, Image 1. In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). Residuals are presented for AOS1 results for Image 1 with $\tau = 10^{-2}$, for four different choices of ς . They demonstrate that the convergence for $u(x)$ is dependent on the width of I_ς , and it is possible to use larger time steps with the improved scheme, AOS1.

AOS1, $\delta = 0.01$				CDF, $\delta = 0.1$				CDF, $\delta = 0.01$			
λ	TC	mb	cpu	λ	TC	mb	cpu	λ	TC	mb	cpu
5	0.99	100	0.16	5	0.97	74	15.3	5	0.97	74	49.4
10	0.99	100	0.11	10	0.90	67	9.1	10	0.90	67	31.0
15	0.99	100	0.28	15	0.88	66	6.4	15	0.87	66	18.6
20	0.99	100	0.12	20	0.87	65	7.6	20	0.87	65	32.9
25	0.99	100	0.12	25	0.87	65	6.1	25	0.87	65	23.9
30	0.99	100	0.17	30	0.86	64	6.2	30	0.86	64	22.8

Table 4.2: Test Set 2. AOS1 and CDF Results for Image 1 (128x128). We present values as λ (the fitting function parameter) varies in terms of segmentation quality, TC , a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion δ , cpu . For CDF we test two stopping criteria, $\delta = 0.1, 0.01$, and for AOS1 we test $\delta = 0.01$. Results demonstrate that AOS1 ($\varsigma = 0.05$) converges faster than CDF, and produces better results in terms of TC and mb for a range of λ .

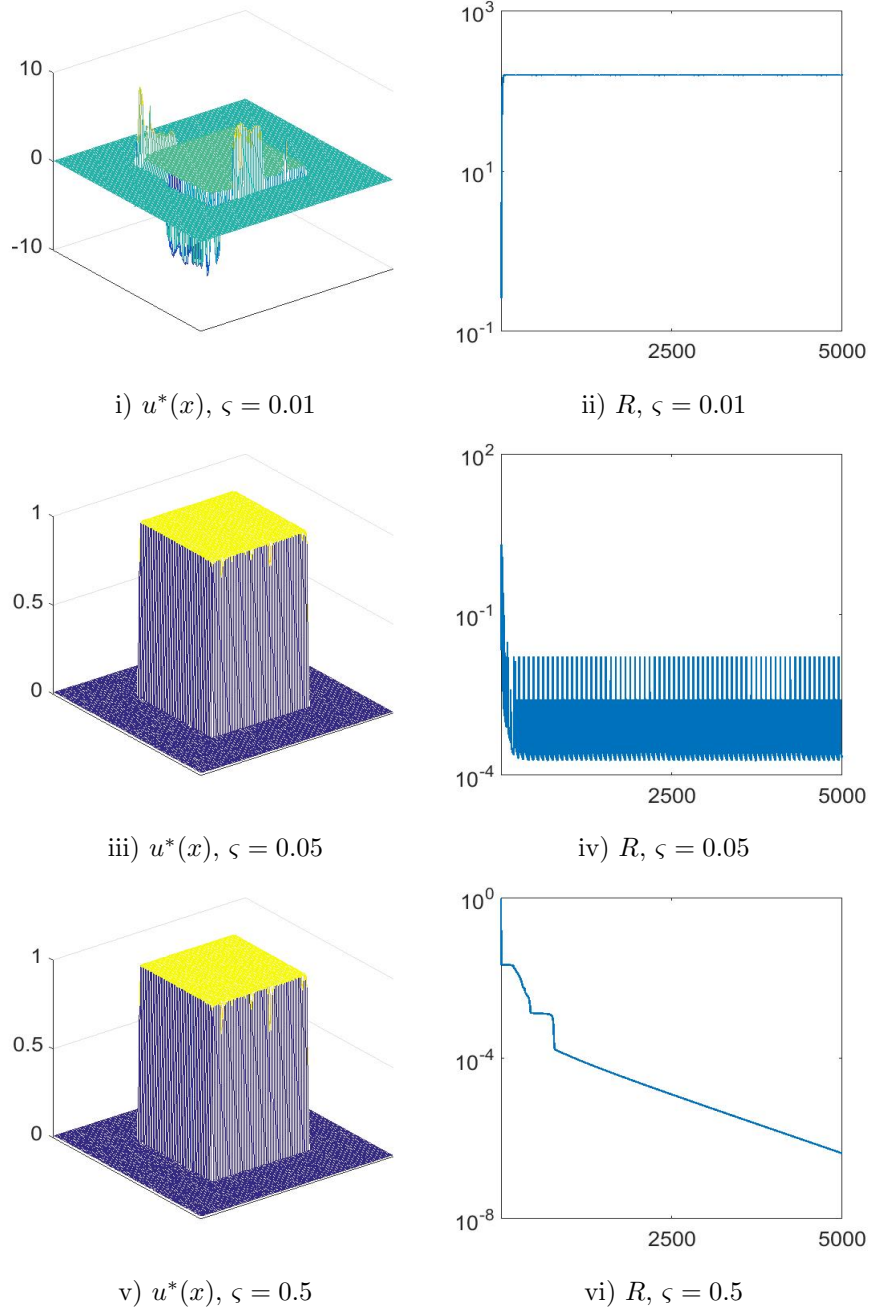
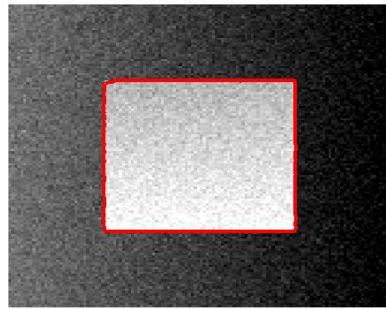
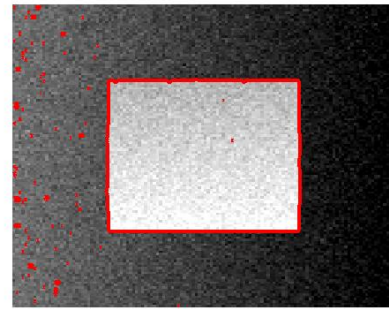


Figure 4.9: Test Set 1. AOS2 ς Results, Image 1. In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). Residuals are presented for AOS2 results for Image 1 with $\tau = 1$, for three different choices of ς . On the left is the segmentation function, $u^*(x)$, and on the right is the residual progression. They demonstrate that the convergence for $u(x)$ is dependent on the width of I_ς , and it is possible to use arbitrary time steps with the improved scheme, AOS2, when $\varsigma = 0.5$.

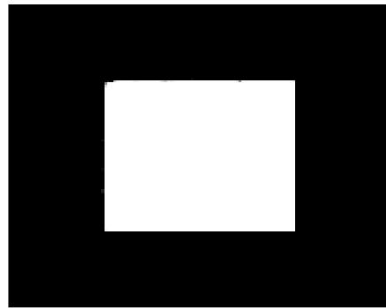
thresholding procedure discussed earlier, does not change. This supports the idea that we are computing global minimisers independently of initialisation. The results demonstrate that I_1 is the most suitable initialisation in terms of *cpu*. As u_0 is initialised further from the ground truth, the computation time can be significantly affected. This drawback can be overcome by selecting ς such that *cpu* is optimal. It is important to



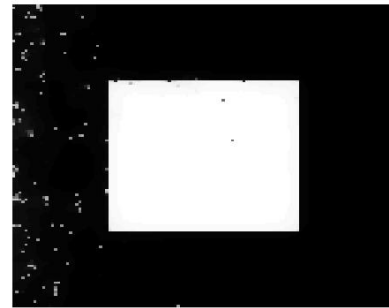
i) Γ^* , AOS2



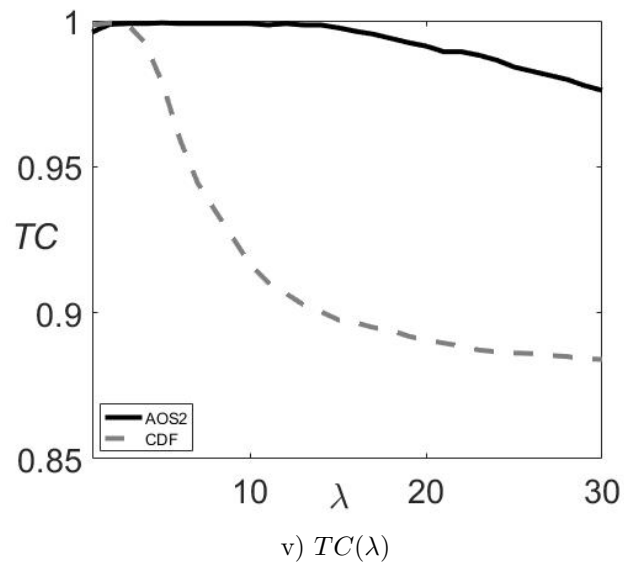
ii) Γ^* , CDF



iii) $u^*(x)$, AOS2



iv) $u^*(x)$, CDF



v) $TC(\lambda)$

Figure 4.10: Test Set 2. Accuracy Results, Image 1. Row 1 is the computed contour Γ^* (given in red on $z(x)$), and the right is the segmentation function $u^*(x)$. On the left are AOS2 results, and the right are CDF results (both for $\lambda = 5$). The plot shows the TC value when λ is varied for AOS2 and CDF. This demonstrates that whilst their best results are similar, AOS2 is successful for a much larger range of the fitting parameter.

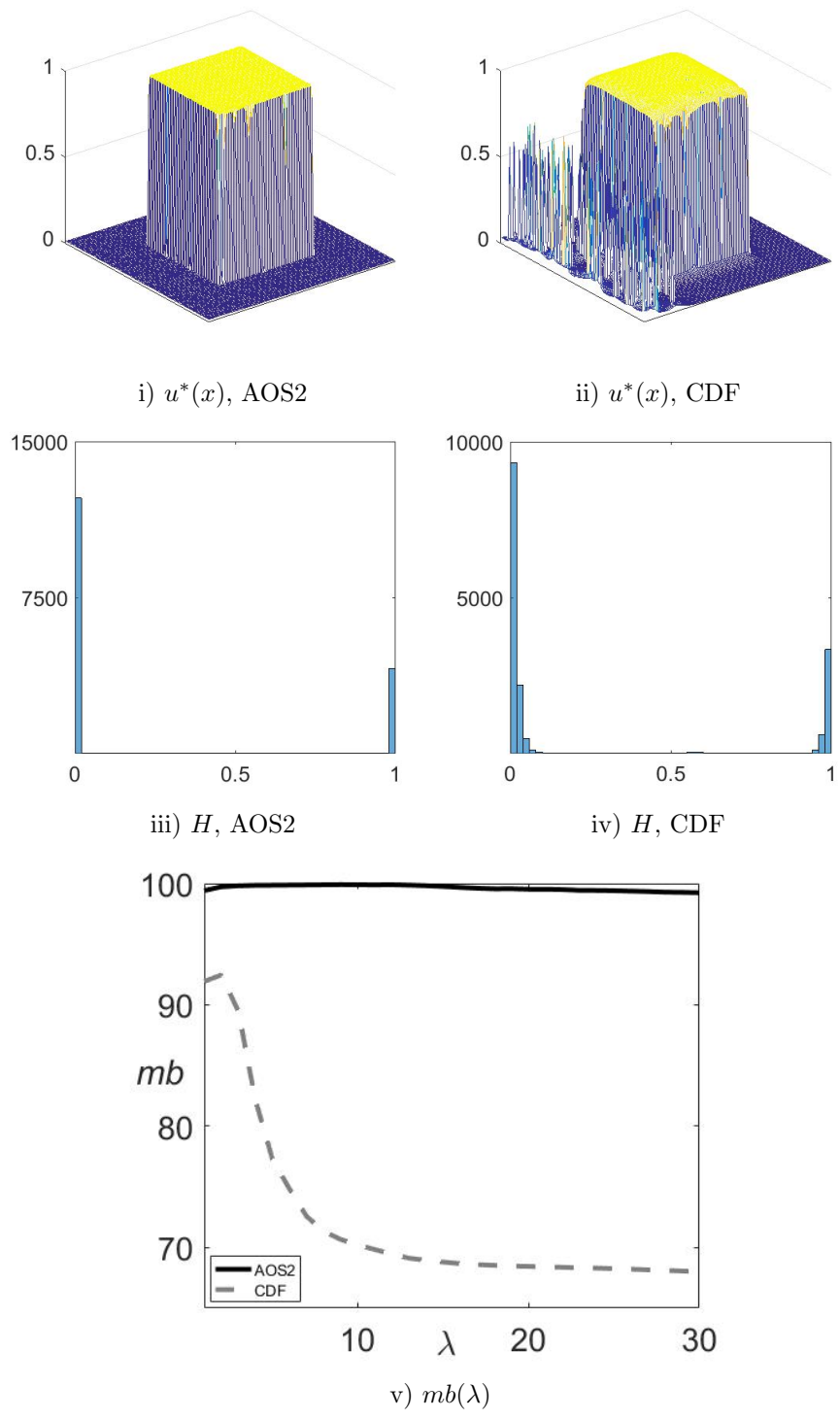


Figure 4.11: Test Set 2. Binary Measurement, Image 1. Row 1 is the segmentation function $u^*(x)$, and row 2 is the histogram for $u^*(x)$. On the left are AOS2 results, and the right are CDF results (both for $\lambda = 5$). The plot shows the mb value when λ is varied for AOS2 and CDF. This demonstrates that AOS2 is consistently closer to a binary result than CDF.

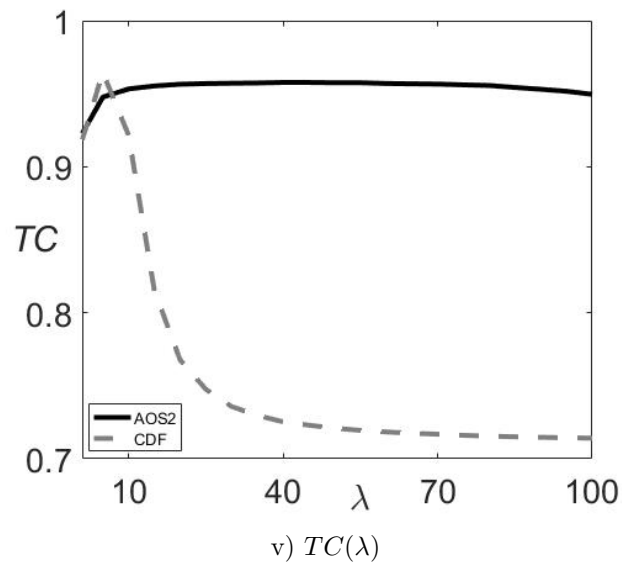
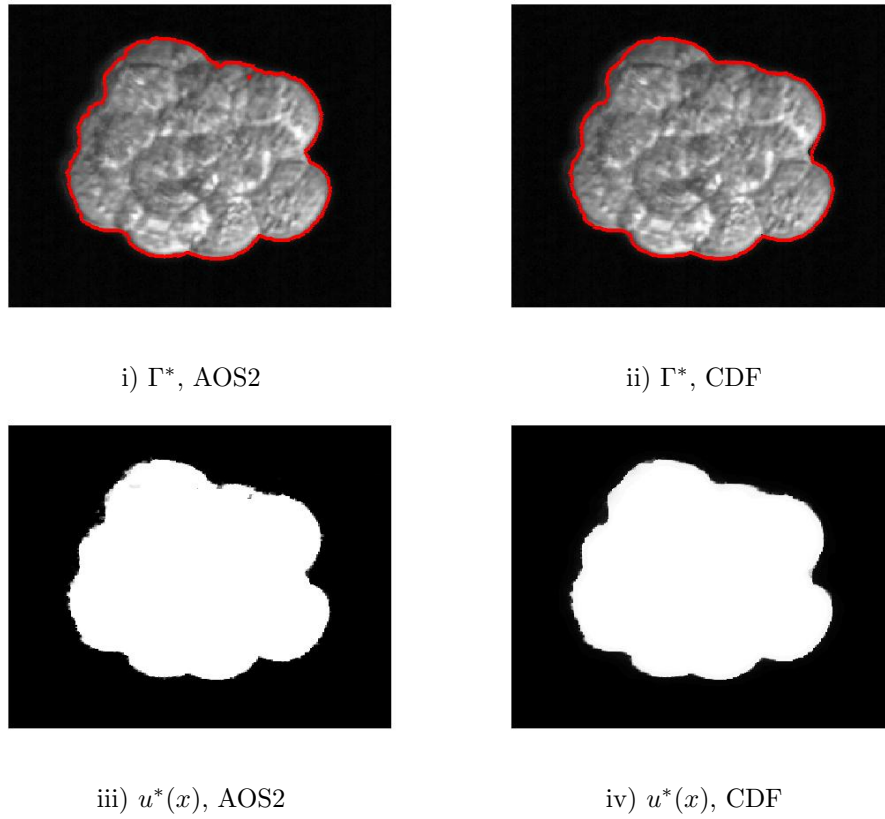


Figure 4.12: Test Set 2. Accuracy Results, Image 2. Row 1 is the computed contour Γ^* (given in red on $z(x)$), and row 2 is the segmentation function $u^*(x)$. On the left are AOS2 results, and the right are CDF results (both for $\lambda = 5$). The plot shows the TC value when λ is varied for AOS2 and CDF. This demonstrates that whilst their best results are similar, AOS2 is successful for a much larger range of the fitting parameter.

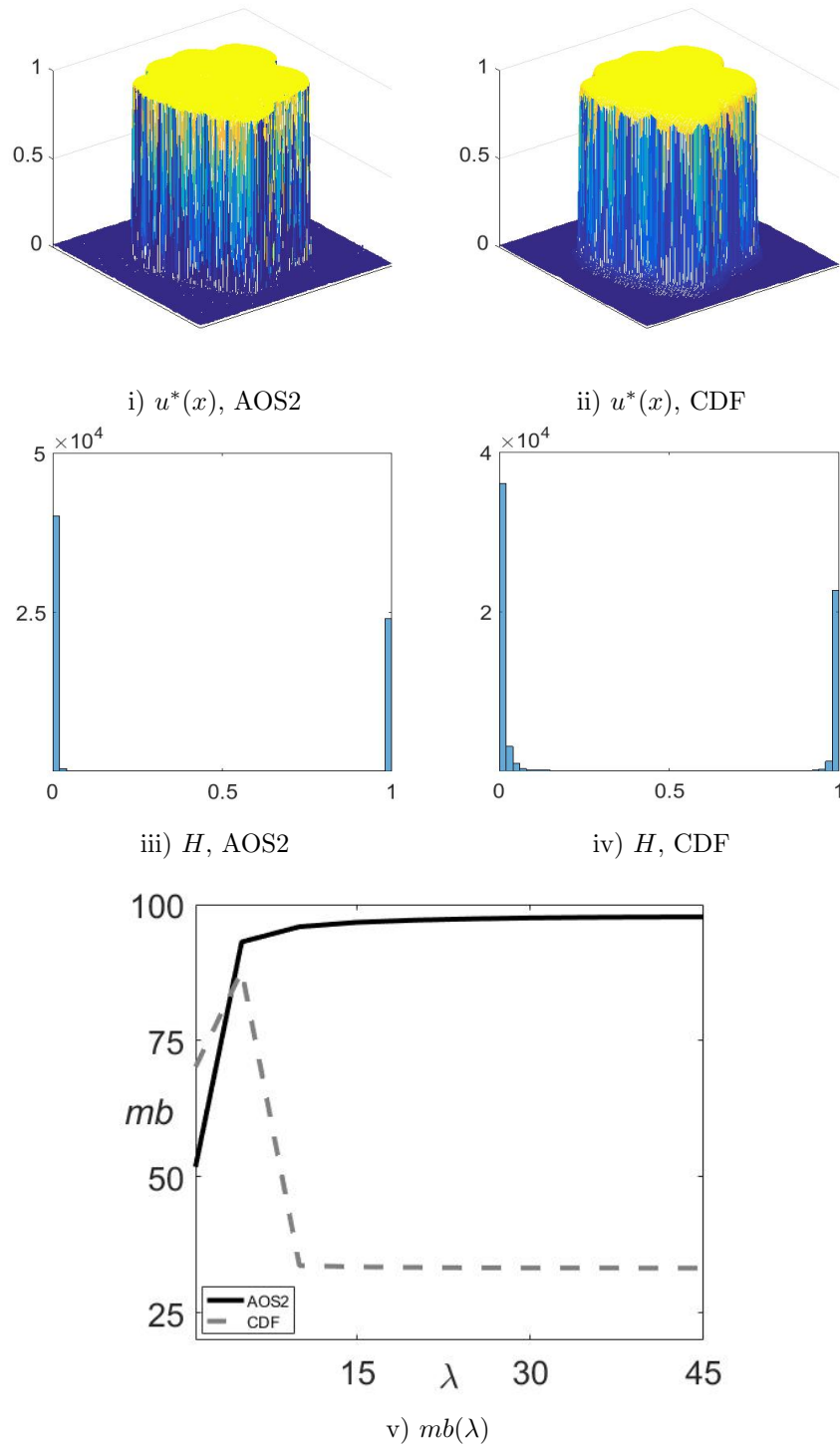


Figure 4.13: Test Set 2. Binary Measurement, Image 2. Row 1 is the segmentation function $u^*(x)$, and row 2 is the histogram for $u^*(x)$. On the left are AOS2 results, and the right are CDF results (both for $\lambda = 5$). The plot shows the mb value when λ is varied for AOS2 and CDF. This demonstrates that AOS2 is consistently closer to a binary result than CDF.

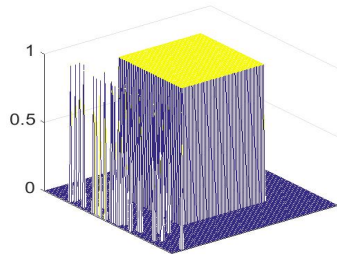
AOS1, $\delta = 0.01$				CDF, $\delta = 0.1$				CDF, $\delta = 0.01$			
λ	TC	mb	cpu	λ	TC	mb	cpu	λ	TC	mb	cpu
5	0.95	94.5	24.8	5	0.96	86	207.1	5	0.96	86	1004.4
10	0.96	96.8	21.1	10	0.94	34	89.9	10	0.94	34	434.2
15	0.96	97.4	21.4	15	0.84	34	52.5	15	0.84	34	261.7
20	0.96	97.6	24.2	20	0.79	34	40.8	20	0.78	34	157.3
25	0.96	97.8	29.5	25	0.76	34	33.6	25	0.76	34	131.5
30	0.96	99.4	29.0	30	0.75	34	33.6	30	0.75	34	137.8

Table 4.3: Test Set 2. AOS1 and CDF Results for Image 2 (256x256). We present values as λ (the fitting function parameter) varies in terms of segmentation quality, TC , a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion δ , cpu . For CDF we test two stopping criteria, $\delta = 0.1, 0.01$, and for AOS1 we test $\delta = 0.01$. Results demonstrate that AOS1 ($\varsigma = 0.05$) converges faster than CDF, and produces better results in terms of TC and mb for a range of λ .

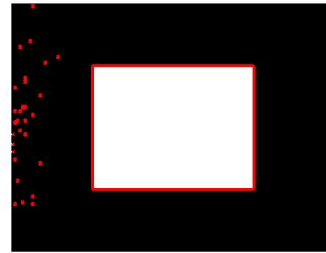
clarify that we cannot provide guidance on the choice of ς , as the behaviour of the scheme is not consistent enough. We tend to set the AOS interval with $\varsigma = 0.05$ in other tests, although for I_4 in Table 4.5 we see that this is not appropriate. It is also important to note that if the stopping criterion is removed, and the iterations are left to run, the results for different initialisations converge to the same solution as we would expect. However, in practice it is important to consider these aspects of the algorithm in order to exploit the improvements in the method. An example of these results is illustrated in Fig. 4.16.

AOS1, mb					AOS1, cpu				
ς	I_1	I_2	I_3	I_4	ς	I_1	I_2	I_3	I_4
0	74	68	70	71	0	319.7	300.1	300.4	301.2
0.001	76	76	74	77	0.001	321.0	304.6	301.9	302.7
0.005	100	99	99	100	0.005	309.1	303.0	303.2	301.4
0.01	100	100	100	100	0.01	1.7	2.2	3.8	4.5
0.05	100	100	100	100	0.05	0.2	3.9	4.5	7.6
0.1	100	99	100	100	0.1	0.08	5.6	6.4	11
0.3	100	99	99	99	0.3	0.2	14.4	14.4	26.7
0.5	100	99	98	99	0.5	0.1	19.2	25.1	43.0

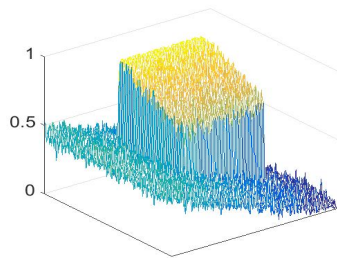
Table 4.4: Test Set 3. Initialisation Results (AOS1), Image 1 (128x128). In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). We present values as ς varies in terms of a measure of how binary $u^*(x)$ is, mb , and the time (in seconds) taken to reach the stopping criterion $\delta = 0.01$, cpu (n.b. for some results the iterations were stopped at the maximum iteration number). Four initialisations are used (shown in Fig. 4.14). Results demonstrate that varying ς affects the convergence time, depending on the choice of initialisation. One notes that whilst $\varsigma = 0.5$ makes convergence likely it can be slower than smaller values.



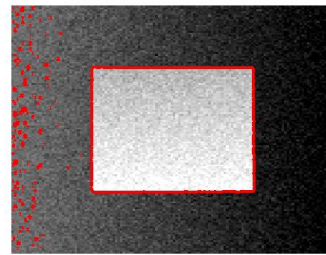
i) $u_0(x), I_1$



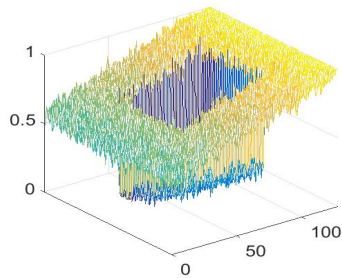
ii) Γ_0, I_1



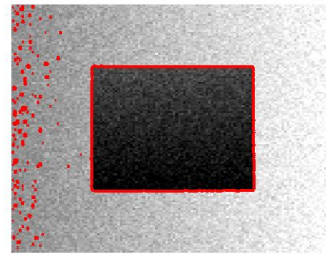
iii) $u_0(x), I_2$



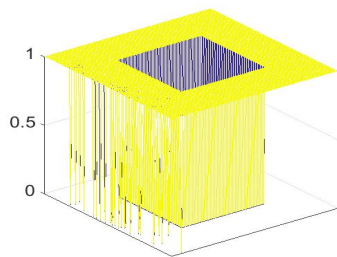
iv) Γ_0, I_2



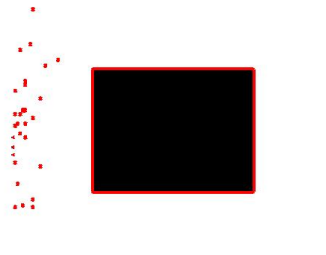
v) $u_0(x), I_3$



vi) Γ_0, I_3

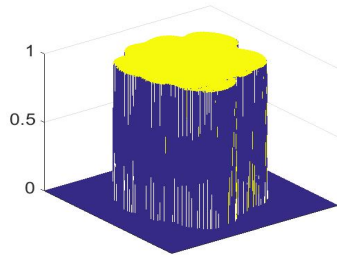


vii) $u_0(x), I_4$

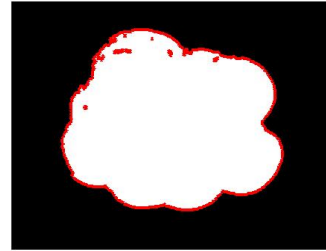


viii) Γ_0, I_4

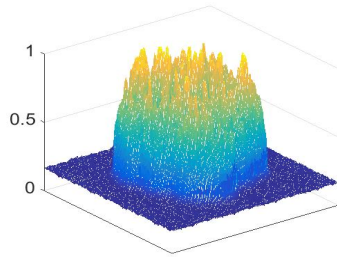
Figure 4.14: Test Set 3. Initialisations, Image 1. Rows 1-4 are for initialisations I_1 - I_4 respectively. On the left is the initial segmentation function $u_0(x)$, and on the right is the initial contour Γ_0 on $u_0(x)$ in red.



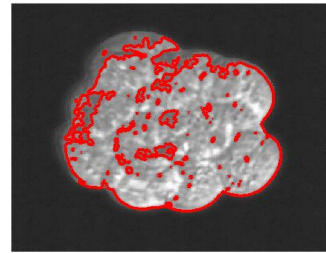
i) $u_0(x), I_1$



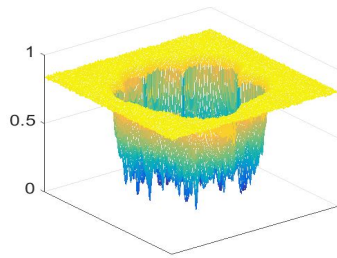
ii) Γ_0, I_1



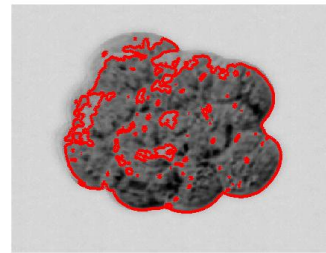
iii) $u_0(x), I_2$



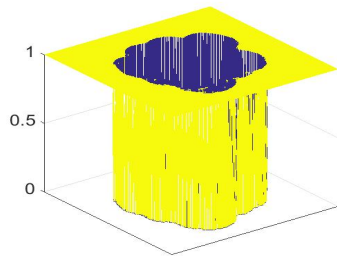
iv) Γ_0, I_2



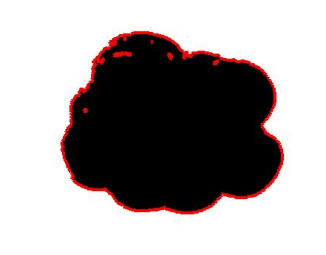
v) $u_0(x), I_3$



vi) Γ_0, I_3



vii) $u_0(x), I_4$



viii) Γ_0, I_4

Figure 4.15: Test Set 3. Initialisations, Image 2. Rows 1-4 are for initialisations I_1 - I_4 respectively. On the left is the initial segmentation function $u_0(x)$, and on the right is the initial contour Γ_0 on $u_0(x)$ in red.

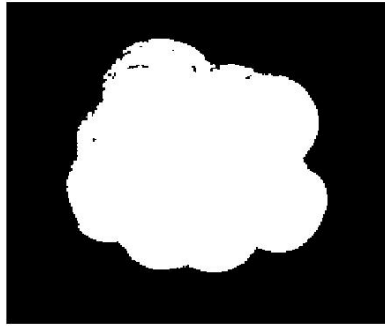
AOS1, <i>mb</i>					AOS1, <i>cpu</i>				
ς	I_1	I_2	I_3	I_4	ς	I_1	I_2	I_3	I_4
0	68	68	68	68	0	1187.7	1188.9	1187.6	1185.8
0.001	69	69	69	69	0.001	1192.5	1188.2	1184.6	1183.9
0.005	74	74	74	74	0.005	1194.9	1197.8	1199.6	1187.9
0.01	97	94	94	95	0.01	14.6	1180.0	1177.8	1184.4
0.05	99	94	98	98	0.05	3.3	28.8	1182.1	1180.9
0.1	99	92	92	95	0.1	3.4	59.5	73.6	144.5
0.3	99	87	87	87	0.3	3.5	99.5	27.8	370.1
0.5	99	85	85	85	0.5	3.3	98.4	52.9	615.8

Table 4.5: Test Set 3. Initialisation Results (AOS1), Image 2 (256x256). In the improved AOS scheme ς determines the width of the interval, I_ς (4.16). We present values as ς varies in terms of a measure of how binary $u^*(x)$ is, *mb*, and the time (in seconds) taken to reach the stopping criterion $\delta = 0.01$, *cpu* (n.b. for some results the iterations were stopped at the maximum iteration number). Four initialisations are used (shown in Fig. 4.15). Results demonstrate that varying ς affects the convergence time, depending on the choice of initialisation. One notes that the best choice of ς is not consistent for different initialisations.

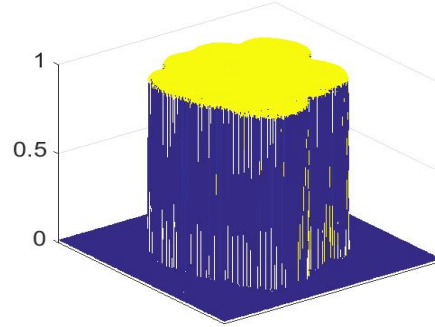
4.6 Remarks

In this chapter, we have proposed a new method to compute global minimisers of two-phase segmentation problems with a generalised fitting function. We have tested our approach against a well-known analogous method [23, 18], i.e. the relaxed binary constraint is imposed with a penalty function, that is widely used in the literature [29, 17, 30, 18]. Our approach compares favourably with this method, with encouraging results in terms of accuracy, computation time, and how close to binary the solution is. Crucially, we also demonstrate that our method appears to be less dependent on the fitting parameter, λ . This is a significant finding in the sense that in practice it makes our method more reliable.

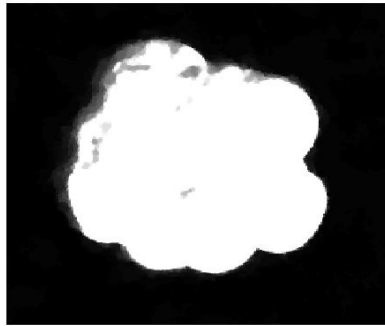
Future work could consist of investigating how the AOS interval, I_ς can be defined in a consistent way. For our second scheme, AOS2, we advise selecting $\varsigma = 0.5$ in line with the theory of Weickert et al. [128] to ensure stable convergence. However, this approach does not reduce computation time significantly in comparison to similar methods. Improvements in this sense are available with AOS1 but depend on the choice of ς . Our work suggests that the choice of initialisation and fitting function contribute to this choice, but it is difficult to provide specific guidance. Further theoretical work on the proposed AOS schemes could help resolve this problem. Also, proving the convexity of the functional for any $\epsilon > 0$ could be explored (as discussed in Section 4.3.2). Currently we can only assert that global minimisers can be computed in practice, but extending this to the general case is desirable.



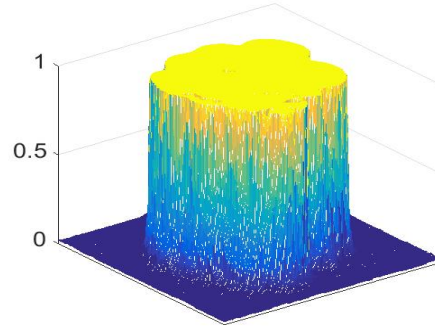
i) $\varsigma = 0.5, I_1$



ii) $\varsigma = 0.5, I_1$



iii) $\varsigma = 0.5, I_2$



iv) $\varsigma = 0.5, I_2$

Figure 4.16: Test Set 3. Initialisation Results. AOS1 results from Table 4.5 with $\delta = 0.01$. The top row is different views of the segmentation function $u^*(x)$ for $\varsigma = 0.5$ and initialisation I_1 . Here $mb = 99$ and $cpu = 3.3$. The bottom row is similar for initialisation I_2 . Here $mb = 85$ and $cpu = 98.4$. This demonstrates that initialising the segmentation function as close to the final result as possible offers significant advantages, both in terms of time it takes to reach the stopping criterion and how close to binary the result is when that happens. This is an example of the observations that can be made from Tables 4.4 and 4.5.

Chapter 5

Global Minimisers of Selective Segmentation Models

5.1 Introduction

We now look at how fitting functions are determined in practice, having introduced globally convex segmentation (GCS) in general in the previous chapter. We will apply these ideas in the context of selective segmentation. Previously, selective segmentation models tended to be level set based and thus finding global minimisers is problematic as discussed in chapters 3 and 4. In this chapter, we discuss the necessary conditions for selective segmentation models to be reformulated in such a way that global minimisers can be found. First, we will clarify the distinction between global and selective segmentation.

Global segmentation is the task of selecting all objects in an image based on a certain characteristic, e.g. intensity, and has been widely studied over the last twenty years [32, 87]. Selective segmentation is when only one object, from within all objects, is selected [8, 104]. With variational segmentation techniques, two main ideas have developed: edge-based methods and region-based methods. As discussed in chapter 3, an important region-based method, where the idea is to achieve segmentation through an approximation of the original image, is the Mumford-Shah functional minimisation [89]; there exists a large literature extending this work [123, 34, 100, 30]. Edge-based methods drive an evolving contour towards edges within an image using an edge detector function. This method was originally proposed by Kass et al. [72]; further work by Caselles et al. led to the Geodesic Active Contours model [22] which is discussed in more depth in chapter 3. Recently, in order to incorporate the advantages of each idea, there has been a tendency to combine edge-based and region-based approaches [83, 22].

The requirements for a selective segmentation model, due to the potential applications, are that solutions are computed quickly and they are reliable with minimal user input. Much research has been done in recent years on developing this idea. In 2005, Gout, Le Guyader and Vese [61] introduced geometrical constraints to Geodesic Active Contours similar to [22] in the form of a set of points on the contour of interest. This idea was enhanced further by Badshah and Chen [8] in 2009, by combining this work with the region-based idea of Mumford-Shah [89] and Chan-Vese [33]. In 2011, to

increase model reliability, Rada et al. [104] introduced a novel Dual Level Set Model, where a local level set incorporates geometrical constraints similar to [61] and [8], locating an object within a global level set. The selective model discussed in detail here is the Rada-Chen model [105], introduced in 2012 to improve on [104] by using a single level set function, where there is a constraint introduced on the area inside the contour. This has proven to be the most effective model [114]. Another idea of improving [8], that is not of the same type as [89], was proposed by Badshah and Chen [9] in 2012, incorporating fitting based on coefficient of variation.

These models, either global or selective, are nonconvex, which can lead to problems in the form of local minima. As previously covered, this means that finding the correct solution is dependent on initialisation, which reduces their reliability. In recent years work has been done to reformulate global segmentation models as convex minimisation problems such that any local minimiser is a global minimiser. The focus of this chapter is to apply the convex reformulation of nonconvex global models to selective segmentation. We remark that related challenges include the idea of selective segmentation based on user input of 3D images. Chan-Vese has been generalised to 3D by Zhang and Chen [139], and user input of a similar type to [8, 105, 61] has been applied with active contours in 3D by Le Guyader and Gout [60]. This involves the selection of points on slices of the 3D data. Visualising objects in this way, allowing for efficient user input, is a difficult problem. In relation to Rada-Chen [105], this input would generate a polyhedron, with its volume providing a selection constraint.

The chapter is organised as follows. In Section 5.2 the idea of global segmentation is discussed, including brief reviews of the work of Mumford-Shah [89], Chan-Vese [33] and Chan, Esedoglu, and Nikolova [30]. We recall this work, first discussed in Chapter 3, in order to provide the full context for the selective reformulation presented later in the chapter. In Section 5.3 selective segmentation is discussed with a review of the most effective model by Rada-Chen [105]. Why this model does not fit in with the convex reformulation idea is explained, motivating the proposal of a new model in Section 5.4. The details of this model are discussed in the nonconvex setting and then reformulated as a convex minimisation problem. Details of the numerical implementation are provided in Section 5.4.4, where we apply the methods introduced for GCS in chapter 4 to this model. Brief details are given for the improved additive operator splitting (AOS) schemes from the previous chapter, which are applicable to this approach by design. Section 5.5 contains results for both the nonconvex and convex models, and we offer some remarks about this work in Section 5.6.

5.2 Global Segmentation

In order to discuss the selective segmentation methods of interest, it is important to introduce global variational image segmentation models. This is important for two reasons; firstly, it will provide the foundation for the selective models introduced and secondly, it provides the method for minimising the associated functionals with the introduction of Active Contours Without Edges [33] by Chan and Vese in 2001.

5.2.1 Piecewise-Constant Mumford-Shah

One of the most important variational image segmentation models is by Mumford and Shah [89], introduced in 1989, and forms the basis for this work as well as many others. Let Ω be a bounded domain in \mathbb{R}^n and z be a bounded measurable function defined on Ω . Here we consider the case where $n = 2$. In the piecewise-constant case, the image, z , is reconstructed as a cartoon of the original where each region, Ω_i , consists of homogeneous intensity (with $i = 1, \dots, N$), separated by an edge set Γ , a closed subset of Ω .

In 2001, Chan and Vese [33] introduced a particular case of the piecewise-constant Mumford-Shah functional. This was the two-phase example ($N = 2$), with $\Omega_1 = \text{in}(\Gamma)$ and $\Omega_2 = \text{out}(\Gamma)$, which looks for the best approximation of an image z by a function u taking only 2 values,

$$u = \begin{cases} c_1 = \text{average of } z \text{ inside } \Gamma, \\ c_2 = \text{average of } z \text{ outside } \Gamma. \end{cases}$$

The length of the set Γ is given by

$$|\Gamma| = \int_{\Gamma} ds.$$

The piecewise-constant two-phase Mumford-Shah functional is given as follows:

$$PC(\Gamma, c_1, c_2) = |\Gamma| + \lambda \int_{\text{in}(\Gamma)} (z - c_1)^2 dx + \lambda \int_{\text{out}(\Gamma)} (z - c_2)^2 dx. \quad (5.1)$$

It consists of the regularisation term, $|\Gamma|$, forcing the boundary between homogeneous regions to be as short and as smooth as possible, and the fitting terms which force the boundary to find regions of homogeneous intensity. Theoretical existence and regularity of minimisers of the piecewise-constant case (5.1), with respect to Γ , are discussed in [89]. However, minimising PC (5.1) is problematic due to the difficulty of tracking the movement of Γ and the model was not implemented directly until the work of [33], as discussed in Chapter 3.

5.2.2 Two-Phase Chan-Vese

Active Contours Without Edges [33], by Chan and Vese, is an important milestone in variational image segmentation. They applied the level set method to minimise the two-phase piecewise-constant Mumford-Shah functional, eqn. (5.1), and overcame the problematic tracking of Γ . Chan and Vese proposed to replace the unknown 1-D variable with a higher dimensional variable, counterintuitively simplifying the problem. They applied the level set method [95], introduced by Osher and Sethian in 1988, to (5.1). By tracking a variable of a higher dimension, where the boundary is represented by a level set of this variable, topological changes in the boundary, such as splitting into two or merging into one, are dealt with automatically. Formally, the boundary Γ is represented by the zero level set of the Lipschitz function ϕ such that

$$\begin{cases} \Gamma = \{(x) \in \Omega \mid \phi(x) = 0\}, \\ \text{in}(\Gamma) = \{(x) \in \Omega \mid \phi(x) > 0\}, \\ \text{out}(\Gamma) = \{(x) \in \Omega \mid \phi(x) < 0\}. \end{cases}$$

The *PC* functional (5.1) is reformulated using the Heaviside function H and the Dirac delta δ defined by

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

In order to compute the associated Euler-Lagrange equation for ϕ we consider regularised versions of H and δ , given as

$$H_\epsilon(\phi) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \frac{\phi}{\epsilon} \right), \quad \delta_\epsilon(\phi) = \frac{1}{\epsilon\pi(1 + \phi^2/\epsilon^2)}.$$

The *PC* functional (5.1) is then reformulated as follows:

$$\begin{aligned} CV(\phi, c_1, c_2) = \int_{\Omega} \delta(\phi) |\nabla H_\epsilon(\phi)| dx + \lambda \int_{\Omega} (z - c_1)^2 H_\epsilon(\phi) dx \\ + \lambda \int_{\Omega} (z - c_2)^2 (1 - H_\epsilon(\phi)) dx, \end{aligned} \quad (5.2)$$

where $\phi(x)$ has been replaced with ϕ for simplicity; this notation will be continued from here. Minimising (5.2) with respect to the intensity constants c_1 and c_2 is given by:

$$c_1(\phi) = \frac{\int_{\Omega} H_\epsilon(\phi) z dx}{\int_{\Omega} H_\epsilon(\phi) dx}, \quad c_2(\phi) = \frac{\int_{\Omega} (1 - H_\epsilon(\phi)) z dx}{\int_{\Omega} (1 - H_\epsilon(\phi)) dx}. \quad (5.3)$$

Then, given these constants, (5.2) is minimised with respect to ϕ :

$$\min_{\phi} CV(\phi, c_1, c_2) \quad (5.4)$$

This leads to the Euler-Lagrange equation

$$\begin{cases} \delta_\epsilon(\phi) \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda \delta_\epsilon(\phi) \left((z - c_1)^2 - (z - c_2)^2 \right) = 0 & \text{in } \Omega, \\ \frac{\partial \phi}{\partial \bar{n}} = 0 & \text{on } \partial \Omega. \end{cases}$$

The work of Chan and Vese is important to the consideration of selective segmentation as it provides the method to tackle segmentation problems of this type in an efficient way. It does however have a drawback in that it involves minimising a nonconvex functional (5.2) over characteristic functions. This means that there are local minima and a computed solution may not be correct unless the initial guess is sufficiently close to the true solution. Fortunately, by reformulating as the minimisation of a convex functional, global minimisers of the nonconvex problem (5.4) can be found. This idea has not yet been applied to selective segmentation models, which also have local minima.

5.2.3 A Global Convex Reformulation

Important to the idea of reformulating a model to be convex is why this improves the reliability of a solution. With that in mind, the fundamental idea behind convex minimisation is now discussed briefly in a general sense. Consider the problem of minimising $f(x)$ subject to $x \in S$, given a non-empty set S . A point $x \in S$ is called a feasible solution to the problem. If $\bar{x} \in S$ and $f(x) \geq f(\bar{x})$ for each $x \in S$, then \bar{x} is a global minimum. If $\bar{x} \in S$ and there exists an ϵ -neighbourhood $N_\epsilon(\bar{x})$ around \bar{x} such that $f(x) \geq f(\bar{x})$ for each $x \in S \cap N_\epsilon(\bar{x})$, then \bar{x} is called a local minimum.

The advantage of convex minimisation is that supposing \bar{x} is a local minimum, if f is convex and S is a convex set, then \bar{x} is a global minimum. It has been shown that minimising the two-phase piecewise-constant Mumford-Shah functional with respect to Γ can be reformulated as a convex problem, by relaxation of the label set. We now introduce the theory behind reformulating the functional (5.1), which we shall later apply to selective segmentation.

We consider the minimisation of the two-phase piecewise-constant Mumford-Shah functional from (5.1) with respect to Γ ; reformulated to the minimisation problem (5.4) by Chan and Vese [33]. Observe that

$$CV(\phi, c_1, c_2) = \int_{\Omega} \delta(\phi) |\nabla H(\phi)| dx + \lambda \int_{\Omega} ((z - c_1)^2 H(\phi) + (z - c_2)^2 (1 - H(\phi))) dx$$

is nonconvex due to the presence of $H(\phi)$. In 2006, Chan, Esedoglu and Nikolova [30] proposed replacing $H(\phi)$ with $u \in [0, 1]$ in (5.2), and obtained the following equivalent, convex, and constrained minimisation problem:

$$\min_{0 \leq u \leq 1} \left\{ \int_{\Omega} |\nabla u| dx + \lambda \int_{\Omega} ((z - c_1)^2 - (z - c_2)^2) u dx \right\}. \quad (5.5)$$

For any fixed $c_1, c_2 \in \mathbb{R}^+$, a global minimiser for $CV(\cdot, c_1, c_2)$ can be found by carrying out the convex minimisation (5.5) [30]. Once the solution u is obtained, set $\Sigma(\gamma) = \{x : u(x) > \gamma\}$ for $\gamma \in (0, 1)$ and then in terms of piecewise-constant two-phase Mumford-Shah, $\Gamma = \partial\Sigma$. As remarked, the convex problem (5.5) will find a global minimiser independently of the initial guess for u .

5.3 The Selective Segmentation Problem and Recent Models

The task of extracting only one object from an image is a challenging problem within segmentation with applications in a number of areas, such as automated object detection in security monitoring and feature selection in medical imaging. Within medical applications, advances in this subject can improve quantitative diagnosis, help monitor treatment over time and improve pre-operative planning.

Here, on image z , we assume the availability of $n_1 (\geq 3)$ points inside the target object that form a set $\mathcal{A} = \{x_i \in \Omega, 1 \leq i \leq n_1\}$ that defines a polygon. A common misconception is that if \mathcal{A} is available any global, nonconvex model (such as [33]) can

solve the selective segmentation problem if one places the initial contour of ϕ near \mathcal{A} . Indeed, this is true for some simple and designed images where features in an image are distinct, but in general this idea does not lead to a useful method for selective segmentation. We also remark that our problem setting is not the same as that of using seeds for fuzzy membership approaches [91, 144]. One model recently proposed by Nguyen et al. [92] attempts another kind of selective segmentation in a similar way and works with a marker set \mathcal{A} and another ‘anti-marker’ set \mathcal{B} which contains points not within the object to be extracted. It uses an edge detector and a probability map, based on user input, but its results tend to be too dependent on user input [114].

In order for a selective method to be suitable in this context, it is imperative that a model requires minimal user input and is reliable. Recent developments in the subject include Gout et al. [61], Badshah-Chen [8] and Rada et al. [104], which include region, edge and geometrical constraints. The geometrical constraints are used to modify the regularisation term by a distance function, $d(x)$, such as the choice used in [8]. It is also possible to alter the regularisation term with the addition of an edge detector (as in [22]), where the strength of detection is adjusted by a parameter, β :

$$g(|\nabla z|) = \frac{1}{1 + \beta|\nabla z|^2}. \quad (5.6)$$

These additions modify the regularisation term [104, 8] to be:

$$\int_{\Gamma} d \cdot g \, ds.$$

Of the selective models studied, two effective models capable of segmenting a wide range of examples in a robust way are by Rada-Chen [105] (based on area constraints) and Badshah-Chen [9] (based on non- L^2 fitting). Here ‘robust’ means that correct segmentations have been obtained as long as the initial contour is strictly inside the object to be extracted.

As with Chan-Vese, these selective models are nonconvex. This means that the models can find local minima, depending on the initialisation of the contour (which are associated with initial contours not strictly within the object to be extracted). This lack of convexity is problematic for a selective segmentation model as reliability and consistency are key in possible applications.

Our intention is to introduce a new nonconvex selective model and reformulate it as a convex minimisation problem, in order to compute the original model’s global minimiser. Our candidates are Rada-Chen [105] and Badshah-Chen [9]. The fitting terms of [9] are based on the coefficient of variation rather than the mean intensity, used in [89, 33]. The convex reformulation idea from Chan et al. [30] was applied to mean intensity fitting terms, so we intend to focus on Rada-Chen [105] (which also uses mean intensity). Also, the geometrical constraints used in [9] can sometimes be too weak based on simple user input, whereas Rada-Chen [105] is less sensitive to the choice of \mathcal{A} . The area constraint of Rada-Chen [105] is an addition to Chan-Vese [33], but is also unsuitable for the convex reformulation. We intend to discuss the reasons

for the lack of suitability in further detail. We provide important details of Rada-Chen [105] below, to demonstrate why the convex reformulation fails here.

From the polygon formed by the marker set \mathcal{A} , denote by A_1 and A_2 respectively the area inside and outside the polygon. The Rada-Chen model [105] makes use of A_1 and A_2 to achieve selective segmentation. The initial contour starts from a polygon inside the object and the additional terms restrict the area inside Γ from growing larger than the target object (and therefore outside the object boundary). It also incorporates the edge detector (5.6) into the regularisation term. We denote the weighted regularisation term as

$$|\Gamma|_g = \int_{\Gamma} g(|\nabla z|) ds.$$

These additions to the two-phase piecewise-constant Mumford-Shah functional (5.1) give us the following energy for selective segmentation:

$$\begin{aligned} RC(\Gamma, c_1, c_2) = & |\Gamma|_g + \lambda \int_{\text{in}(\Gamma)} (z - c_1)^2 dx + \lambda \int_{\text{out}(\Gamma)} (z - c_2)^2 dx \\ & + \frac{\theta}{2} \left[\left(\int_{\text{in}(\Gamma)} d\xi - A_1 \right)^2 + \left(\int_{\text{out}(\Gamma)} d\xi - A_2 \right)^2 \right]. \end{aligned} \quad (5.7)$$

Using the level set formulation, this energy (5.7) becomes [105]:

$$\begin{aligned} RC(\phi, c_1, c_2) = & \int_{\Omega} g(|\nabla z|) \delta(\phi) |\nabla H(\phi)| dx \\ & + \lambda \int_{\Omega} (z - c_1)^2 H(\phi) dx + \lambda \int_{\Omega} (z - c_2)^2 (1 - H(\phi)) dx \\ & + \frac{\theta}{2} \left[\left(\int_{\Omega} H(\phi) d\xi - A_1 \right)^2 + \left(\int_{\Omega} (1 - H(\phi)) d\xi - A_2 \right)^2 \right]. \end{aligned} \quad (5.8)$$

The energy is minimised successively with respect to the intensity constants, c_1 and c_2 given by (5.3), and ϕ . The nonconvex problem of minimising (5.8) with respect to ϕ ,

$$\min_{\phi} RC(\phi, c_1, c_2) \quad (5.9)$$

leads to the Euler-Lagrange equation, where $g = g(|\nabla z|)$,

$$\begin{cases} \delta_{\epsilon}(\phi) \nabla \cdot \left(g \frac{\nabla \phi}{|\nabla \phi|} \right) + f = 0 & \text{in } \Omega, \\ \frac{\partial \phi}{\partial \mathbf{n}} = 0 & \text{on } \partial \Omega, \end{cases} \quad (5.10)$$

and

$$\begin{aligned} f = & -\lambda \delta_{\epsilon}(\phi) \left\{ (z - c_1)^2 - (z - c_2)^2 \right\} \\ & - \theta \delta_{\epsilon}(\phi) \left\{ \left(\int_{\Omega} H_{\epsilon}(\phi) dx - A_1 \right) - \left(\int_{\Omega} (1 - H_{\epsilon}(\phi)) dx - A_2 \right) \right\}. \end{aligned}$$

Solving (5.10) can be done with the introduction of an artificial time step and using the gradient descent method, as discussed in Chapter 2:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \nabla \cdot \left(g \frac{\nabla \phi}{|\nabla \phi|} \right) + f.$$

We now discuss the possibility of reformulating (5.9) into a convex minimisation problem. There are two reasons which mean this is not possible, which have to be considered for the proposal of an appropriate model. Firstly, the additional terms, based on A_1 and A_2 , only incorporate the area of the object into the functional (5.8). This means that information about the location of the object is provided by the initialisation. Clearly, convex reformulation where a global minimiser is found independently of initialisation is not feasible in this case. Secondly, the method of convex reformulation of Chan et al. [30] introduced above requires linearity in $H(\phi)$, in the fitting term of (5.8). The area constraint of Rada-Chen [105] violates this condition. This provides the two main considerations in proposing a new selective model, suitable for convex reformulation, which we detail next.

5.4 Proposed Distance Selective Segmentation Model

In the following is the introduction of our new model that fits in with the idea of being reformulated as a convex minimisation problem and is broadly speaking analogous to Rada-Chen [105]. It uses the same user input as [105], whilst instead of penalising the area inside the contour from growing too much, it penalises the contour from moving further away from the polygon, a set of points denoted by \mathcal{P} , formed by the user input set, \mathcal{A} . The new constraint is linear in the indicator function and includes locational information of the target object, consistent with the idea of convex reformulation.

5.4.1 A New Nonconvex Selective Model

The proposed nonconvex model, to be called Distance Selective Segmentation (DSS), has a different area fitting term than Rada-Chen [105]. Here, $P_0(x)$ is Euclidean distance of each point $x \in \Omega$ from its nearest point in the polygon, made up of $(x_p) \in \mathcal{P}$, constructed from the user input set, \mathcal{A} . Then the function $P_d(x)$ is given as:

$$P_d(x) = \frac{P_0(x)}{\|P_0\|_{L^\infty}}. \quad (5.11)$$

The DSS functional is then defined as:

$$\begin{aligned} DSS(\Gamma, c_1, c_2) = & |\Gamma|_g + \theta \int_{\text{in}(\Gamma)} P_d(x) dx \\ & + \lambda \int_{\text{in}(\Gamma)} (z - c_1)^2 dx + \lambda \int_{\text{out}(\Gamma)} (z - c_2)^2 dx. \end{aligned} \quad (5.12)$$

Here, we have the regularisation and fitting terms from the two-phase piecewise-constant Mumford-Shah functional (5.1) with the addition of a new distance fitting term, nor-

malised so that $P_d(x, y) \in [0, 1]$. For $x \in \mathcal{P}$, $P_d(x) = 0$ and (5.12) reduces to (5.1), except the regularisation term is weighted by an edge detector function (5.6) as in [18, 105]. Introducing the level set formulation, (5.12) reduces to the following model:

$$\min_{\phi, c_1, c_2} \left\{ DSS^{LS}(\phi, c_1, c_2) = \int_{\Omega} \delta_{\epsilon}(\phi) g |\nabla \phi| dx + \theta \int_{\Omega} H_{\epsilon}(\phi) P_d dx \right. \\ \left. + \lambda \int_{\Omega} H_{\epsilon}(\phi) (z - c_1)^2 dx + \lambda \int_{\Omega} (1 - H_{\epsilon}(\phi)) (z - c_2)^2 dx \right\}, \quad (5.13)$$

Here, if the area parameter, θ , is too strong the final result will just be the polygon \mathcal{P} which of course is undesirable. The idea behind the P_d term is that it encourages $H(\phi) \in \Omega \setminus \mathcal{P}$ to be 0, enforced more strictly the further from the object of interest a point is. The motivation behind this new model is that it fits in with the idea of convex reformulation.

But it is important to clarify whether the idea behind this segmentation model, i.e. the distance constraint, works as it is. The answer is yes. Comparisons of (5.13) with Rada-Chen [105] are made for three examples and shown in Figures 5.1-5.2 of Section 5.5.1. There, one clearly observes that the two sets of segmentation results are successful. That is, (5.13) is a valid selective segmentation in its own right. In the third example, where the initial guess is altered, both results are unsuccessful as local minima have been found. We look to correct this fault in DSS (5.13) by convexification of the model.

5.4.2 A Selective Convex Reformulation

We now present details for the convex reformulation of (5.13). As in [30], this energy can be made convex by making the adjustment $H_{\epsilon}(\phi) \rightarrow u \in [0, 1]$ to give the Convex Distance Selective Segmentation (CDSS) functional:

$$CDSS(u, c_1, c_2) = \int_{\Omega} |\nabla u|_g dx + \lambda \int_{\Omega} r u dx + \theta \int_{\Omega} P_d u dx \quad (5.14)$$

where $r = (z - c_1)^2 - (z - c_2)^2$ and $|\nabla u|_g = g(|\nabla z|)|\nabla u|$. Given initial values for c_1 and c_2 , based on the set \mathcal{A} , our model consists of the following constrained minimisation problem:

$$\min_{0 \leq u \leq 1} CDSS(u, c_1, c_2). \quad (5.15)$$

Define $\Sigma(\gamma) = \{x : u(x) \geq \gamma\}$ for $\gamma \in (0, 1)$. Following the work of Chan et al. [30], we can demonstrate that a minimiser for DSS (5.13) is given by (5.15). Using the Coarea formula [57], for the first term, the weighted total variation (TV) norm, in (5.12), we get

$$\begin{aligned}
\int_{\Omega} |\nabla u|_g dx &= \int_0^1 g(|\nabla z|) \text{Per}\left(\{(x, y) : u(x, y) \geq \gamma\}; \Omega\right) d\gamma \\
&= \int_0^1 g(|\nabla z|) \text{Per}\left(\Sigma(\gamma); \Omega\right) d\gamma = \int_0^1 |\Gamma|_g d\gamma.
\end{aligned} \tag{5.16}$$

For the remaining terms in (5.12) we first need to recall a definition from Chapter 2. Let u be a non-negative, real-valued, measurable function on Ω . Then with χ a characteristic function,

$$u(x) = \int_0^\infty \chi_{u(x) > t} dt.$$

For the first fitting term, as $u \in [0, 1]$, we have

$$\begin{aligned}
\int_{\Omega} (z - c_1)^2 u dx &= \int_{\Omega} (z - c_1)^2 \int_0^1 \chi_{\Sigma(\gamma)} d\gamma dx = \int_0^1 \int_{\Omega} (z - c_1)^2 \chi_{\Sigma(\gamma)} dx d\gamma \\
&= \int_0^1 \int_{\Sigma(\gamma)} (z - c_1)^2 dx d\gamma,
\end{aligned} \tag{5.17}$$

and for the other two terms, similarly, we have

$$\int_{\Omega} (z - c_2)^2 u dx = \int_0^1 \int_{\Sigma(\gamma)} (z - c_2)^2 dx d\gamma = C - \int_0^1 \int_{\Omega \setminus \Sigma(\gamma)} (z - c_2)^2 dx d\gamma, \tag{5.18}$$

$$\int_{\Omega} P_d u dx = \int_0^1 \int_{\Sigma(\gamma)} P_d dx d\gamma, \tag{5.19}$$

where $C = \int_{\Omega} (z - c_2)^2 dx$ and is independent of u . Combining equations (5.16)-(5.19):

$$\begin{aligned}
CDSS(u, c_1, c_2) &= \int_0^1 \left\{ |\Gamma|_g + \lambda \int_{\Sigma(\gamma)} (z - c_1)^2 dx \right. \\
&\quad \left. + \lambda \int_{\Omega \setminus \Sigma(\gamma)} (z - c_2)^2 dx + \theta \int_{\Sigma(\gamma)} P_d dx \right\} d\gamma - C \\
&= \int_0^1 DSS(\Gamma, c_1, c_2) d\gamma - C.
\end{aligned}$$

Since C is independent of u , it follows that if u is a minimiser of $CDSS(\cdot, c_1, c_2)$ then for $\gamma \in (0, 1)$ the set $\Gamma = \Sigma(\gamma)$ is a minimiser of $DSS(\cdot, c_1, c_2)$. However, the convex minimisation problem (5.15) will provide us with the ability to find a global minimiser, independently of initialisation.

5.4.3 Unconstrained Minimisation

The constrained minimisation problem (5.15) can be replaced by an unconstrained one:

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

where $\nu(u) = \max\{0, 2|u - 1/2| - 1\}$ is an exact penalty term [65], provided that $\alpha > \frac{1}{2}\|\lambda r + \theta P_d\|_{L^\infty}$ (see a proof in [30] for a related problem). In order to compute the associated Euler-Lagrange equation for u we recall the regularised version of the penalty function, $\nu_\epsilon(u)$, introduced in Chapter 4:

$$\nu_\epsilon(u) = H_\epsilon\left(\sqrt{(2u-1)^2 + \epsilon} - 1\right)\left[\sqrt{(2u-1)^2 + \epsilon} - 1\right],$$

where $H_\epsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi} \arctan \frac{x}{\epsilon}\right)$. Then we get the following Euler-Lagrange equation for u :

$$\begin{cases} \nabla \cdot \left(g \frac{\nabla u}{|\nabla u|}\right) - \lambda r - \theta P_d - \alpha \nu'_\epsilon = 0 & \text{in } \Omega, \\ \frac{\partial u}{\partial \bar{n}} = 0 & \text{on } \partial\Omega. \end{cases} \quad (5.20)$$

To minimise for the intensity values, we use the following equations:

$$c_1(u) = \frac{\int_\Omega uz \, dx}{\int_\Omega u \, dx}, \quad c_2(u) = \frac{\int_\Omega (1-u)z \, dx}{\int_\Omega (1-u) \, dx}. \quad (5.21)$$

5.4.4 Numerical Implementation

Equation (5.20) can be solved by the gradient descent method by solving the following:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{g \nabla u}{|\nabla u|}\right) - \lambda r - \theta P_d - \alpha \nu'_\epsilon. \quad (5.22)$$

It is possible to use an explicit or semi-implicit time marching scheme, which we discount based on our discussion in Chapters 2 and 4. As with Chapter 4 we apply the semi-implicit AOS scheme proposed by [85, 129]. Again, to avoid singularities we replace $|\nabla u|$ with $|\nabla u|_\beta = \sqrt{u_x^2 + u_y^2 + \beta}$ for small β , and denote $W = \frac{g}{|\nabla u|_\beta}$. Freezing W linearises the equation and (5.22) can be rewritten in the form:

$$\frac{\partial u}{\partial t} = \partial_x(W \partial_x u) + \partial_y(W \partial_y u) - \lambda r - \theta P_d - \alpha \nu'_\epsilon$$

We treat the equation in a similar way to Chapter 4 and [112], which we will recall now. In the implementation of this method, we use the improved AOS schemes, AOS1 and AOS2. In terms of the penalty function, based on the results from Chapter 4, we set the regularisation parameter at $\epsilon = 0.01$. Typically, when computation time is a priority we use AOS1 ($\tau = 0.01, \varsigma = 0.1$) and otherwise we use AOS2 ($\tau = 1, \varsigma = 0.5$). In Test Set 3 we vary these parameters to observe the advantages of these schemes where the fitting function is not fixed. We recall the essential details from Chapter 4, for the improved AOS schemes. First we summarise the original scheme, AOS0:

$$u_\ell^{n+1} = \underbrace{(I - 2\tau A_\ell(u^n))^{-1}}_{Q_0} \underbrace{(u^n - \tau \alpha \nu'_\epsilon(u^n) - \tau \lambda f)}_{f_0}.$$

The AOS interval, controlled by ς , is

$$I_\varsigma := [0 - \varsigma, 0 + \varsigma] \cup [1 - \varsigma, 1 + \varsigma], \quad \tilde{b}^n = \begin{cases} b, & \text{if } u^n \in I_\varsigma \\ 0, & \text{elsewhere.} \end{cases}$$

The first proposed AOS scheme from Chapter 4 is summarised as

$$u_\ell^{n+1} = \underbrace{\left(I + \tilde{B}^n - 2\tau A_\ell(u^n) \right)^{-1}}_{Q_1} \underbrace{\left(u^n + \tau \alpha \tilde{b} u^n - \tau \alpha \nu'_\epsilon(u^n) - \tau \lambda f \right)}_{f_1}.$$

The second proposed AOS scheme from Chapter 4 is summarised as

$$u_\ell^{n+1} = \underbrace{\left(I - 2\tau(I + \tilde{B}^n)^{-1} A_\ell(u^n) \right)^{-1}}_{Q_2} \underbrace{\left(u^n - \tau(I + \tilde{B}^n)^{-1} (\alpha \nu'_\epsilon(u^n) + \lambda f) \right)}_{f_2}.$$

Here, A_ℓ is the diffusion quantity in the ℓ direction ($\ell = 1, 2$ for x and y directions respectively) and was derived using the finite difference method, τ is the time step size and n denotes the n^{th} iteration. Details and background associated with this are given in Chapter 2. The matrices A_ℓ are given as follows, where $W_{ij}^n = W(u_{ij}^n)$, and h_x and h_y are the grid sizes in the x and y directions respectively:

$$\begin{aligned} \left(A_1(u^n) u^{n+1} \right)_{i,j} &= \left(\partial_x (W^n \partial_x u^{n+1}) \right)_{i,j} \\ &= \frac{1}{h_x} \left(W_{i+1/2,j}^n (\partial_x u^{n+1})_{i+1/2,j} - W_{i-1/2,j}^n (\partial_x u^{n+1})_{i-1/2,j} \right) \\ &= \frac{1}{h_x} \left(\frac{W_{i+1,j}^n + W_{i,j}^n}{2} \frac{u_{i+1,j}^{n+1} - u_{i,j}^{n+1}}{h_x} - \frac{W_{i,j}^n + W_{i-1,j}^n}{2} \frac{u_{i,j}^{n+1} - u_{i-1,j}^{n+1}}{h_x} \right) \\ &= u_{i+1,j}^{n+1} \left(\frac{W_{i+1,j}^n + W_{i,j}^n}{2h_x^2} \right) + u_{i-1,j}^{n+1} \left(\frac{W_{i-1,j}^n + W_{i,j}^n}{2h_x^2} \right) \\ &\quad - u_{i,j}^{n+1} \left(\frac{W_{i+1,j}^n + W_{i-1,j}^n + 2W_{i,j}^n}{2h_x^2} \right) \end{aligned}$$

and similarly,

$$\begin{aligned} \left(A_2(u^n) u^{n+1} \right)_{i,j} &= \left(\partial_y (W^n \partial_y u^{n+1}) \right)_{i,j} = u_{i,j+1}^{n+1} \frac{W_{i,j+1}^n + W_{i,j}^n}{2h_y^2} \\ &\quad + u_{i,j-1}^{n+1} \frac{W_{i,j-1}^n + W_{i,j}^n}{2h_y^2} - u_{i,j}^{n+1} \left(\frac{W_{i,j+1}^n + W_{i,j-1}^n + 2W_{i,j}^n}{2h_y^2} \right). \end{aligned}$$

As before the update is given as

$$u^{n+1} = \frac{u_1^{n+1} + u_2^{n+1}}{2}.$$

Full details can be given in Chapter 4. Each method (AOS1 and AOS2) demonstrate quantifiable advantages over the original AOS scheme [85, 129] for GCS, and over Chambolle's dual formulation [23, 18].

The New Algorithm

The algorithm computes a solution for a sequence of alternating minimisation problems. For each fixed c_1 and c_2 we have a new minimisation problem, which is solved using AOS0, AOS1, or AOS2. The final solution, when c_1 and c_2 have converged, is denoted u^* . It is worth noting that alternative algorithms can be obtained by fixing c_1 and/or c_2 depending on the problem. If the image is not quite piecewise-constant, varying at least one is advisable. If both are fixed then we can guarantee finding the global minimiser of the functional.

Algorithm 1 AOS method for CDSS

- 1: Set θ . Calculate g and P_d using (5.6) and (5.11) respectively.
 - 2: Initialise $u^{(0)}$ such that Γ is the boundary of \mathcal{P} .
 - 3: **for** $k \leftarrow 1$: *maxit* **do**
 - 4: Calculate $c_1^{(k)}(u^{(k-1)})$ and $c_2^{(k)}(u^{(k-1)})$ using (5.21)
 - 5: Calculate $r_p^{(k)} = \lambda \left((z - c_1^{(k)})^2 - (z - c_2^{(k)})^2 \right) + \theta P_d$.
 - 6: Set $\alpha^{(k)} = \|r_p^{(k)}\|_{L^\infty}$.
 - 7: $u^{(k)} \leftarrow \min_u CDSS \left(c_1^{(k)}, c_2^{(k)}, \alpha^{(k)} \right)$ using AOS scheme.
 - 8: **end for**
 - 9: $u^* \leftarrow u^{(k)}$.
-

5.5 Experimental Results

This section will show three sets of experiments to test the effectiveness of our new algorithms and to compare them with the existing model. In the following we select the parameters as follows. We have found that setting $\epsilon = 10^{-2}$ produces a tight approximation of $\nu(u)$. We fix the penalty parameter at $\alpha = \|\lambda r + \theta P_d\|_{L^\infty}$, which is enough to enforce the constraint [30]. We set the time step at $\tau = 10^{-2}$ and $\varsigma = 0.1$, except in Test Set 3, where they are varied to demonstrate the benefits of the improved AOS method. The only restriction on β is that it is small; we select it as $\beta = 10^{-6}$ [109]. We have to consider the balance between the regularisation and fitting term, which will change for each problem. We vary λ for each problem, depending on the shape and smoothness of the boundary of the desired object. It might be worth considering the work of Mylona et al. [90] who automatically optimise these parameters based on image information. The following tests use only three points input by the user, i.e. $n_1 = 3$. The model is capable of achieving the desired result with a simple shape within the target, even for awkwardly shaped targets as seen in Figs. 5.3 and 5.4. The resilience to these selections is discussed further in 5.5.2. This leaves the main choice for a successful segmentation as the distance selection parameter, θ . In these tests, it varies between 1 and 4.5. The basis for this choice is the size of the target object and its proximity to other image features of similar intensity, and can be intuitively selected quite reliably.

In Test Set 1 results are presented for the proposed nonconvex Distance Selective Segmentation (DSS) model and compared to the successful Rada-Chen model [105],

demonstrating its robustness in difficult cases, whilst underlining the need for the convex reformulation. In Test Set 2, results are presented for the Convex Distance Selective Segmentation (CDSS) model, demonstrating its success in segmentation of a range of examples independently of initialisation and its robustness to user input. Test Set 3 demonstrates quantitative improvement of the new AOS method, in relation to one example. All images tested are of size 128x128.

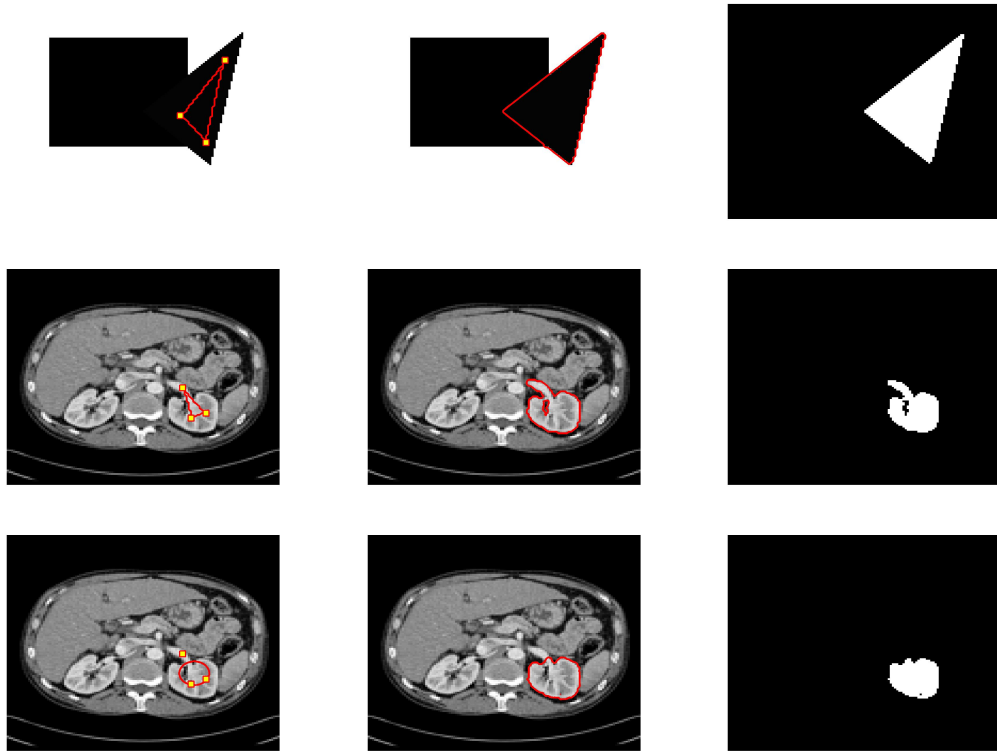


Figure 5.1: Test Set 1. Results for Rada-Chen [105], for three test problems (given by rows 1-3). From left to right: initialisation (with user input set \mathcal{A}), final contour, object selected

5.5.1 Test Set 1 (Nonconvex Model Comparisons)

In Fig. 5.1 results are presented for three examples for Rada-Chen [105] and in Fig. 5.2 the same examples are presented for DSS. Results demonstrate that the new model can also produce the successful results of Rada-Chen [105], whilst both models are sensitive to initialisation, as evident in row 3 of each figure. The nature of the failure in each case is due to finding a local minimum, as is possible for the nonconvex formulation. This is evident from the fact that the user input set, \mathcal{A} , is the same for rows 2 and 3 whilst the initialisations are different, and one case fails where as the other succeeds. This provides the motivation for convexifying the energy in the DSS case, as this cause of failure is removed.

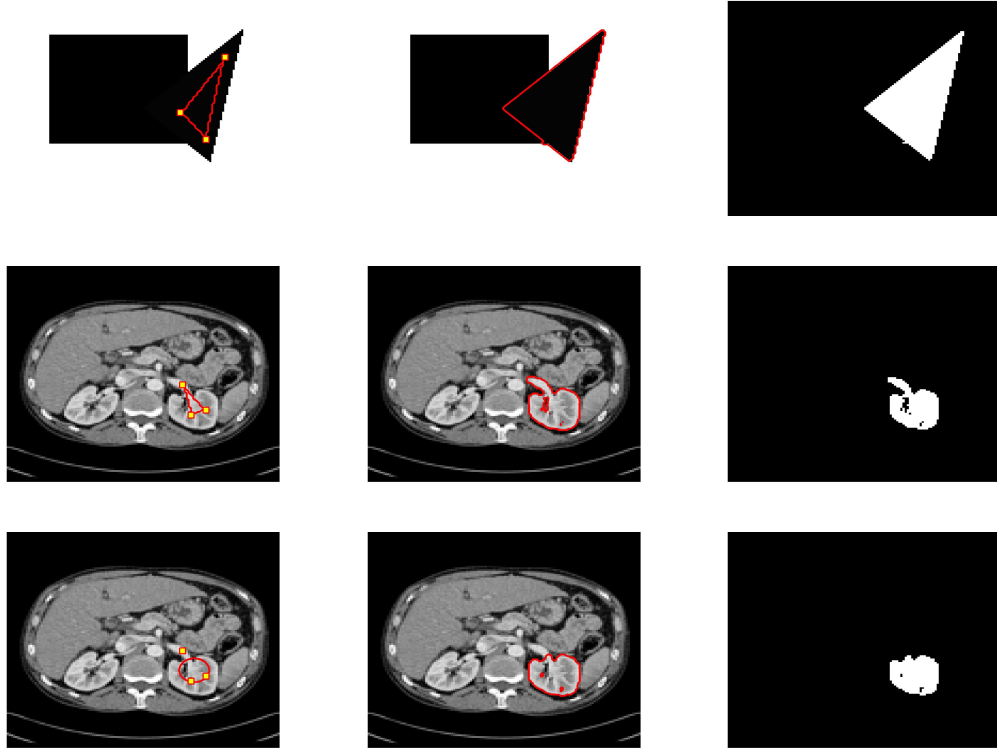


Figure 5.2: Test Set 1. Results for DSS, for three test problems (given by rows 1-3). From left to right: initialisation (with user input set \mathcal{A}), final contour, object selected

5.5.2 Test Set 2 (Robustness to User Selection)

In Fig. 5.3 results for CDSS are presented for three examples. The function is initialised as the given image, with successful segmentation in each case. In Figs. 5.4 and 5.5 the same object is selected, with different user input for each. The solution (ground truth) is given by an ideal user input set, \mathcal{A}^* , which is the shape of the target object and would require n_1 to be large. This is not feasible in practice, as it essentially consists of a manual segmentation. We intend to demonstrate that an acceptable approximation of the solution can be achieved with only three points ($n_1 = 3$), even when segmenting a difficult shape. We have two choices of user input, \mathcal{A}_4 from Fig. 5.4 and \mathcal{A}_5 from Figure 5.5. Whilst \mathcal{A}_5 is close to the boundary of the target (and closer to the ideal user input, \mathcal{A}^*), \mathcal{A}_4 is a more interior selection. These produce slightly different results, but both are acceptable. This demonstrates that even with a simple user input far from the ideal, such as \mathcal{A}_4 , we get an acceptable result. A more appropriate user input (i.e. closer to the ideal), such as \mathcal{A}_5 , produces a better result, but still only requires three points. One observes that the initialisations were deliberately chosen to be not within the object intended (which would fail with all other nonconvex models) and yet CDSS is capable of locating the boundary of the target object. These examples demonstrate the robustness of the model; successful segmentation is possible for a wide range of user input.

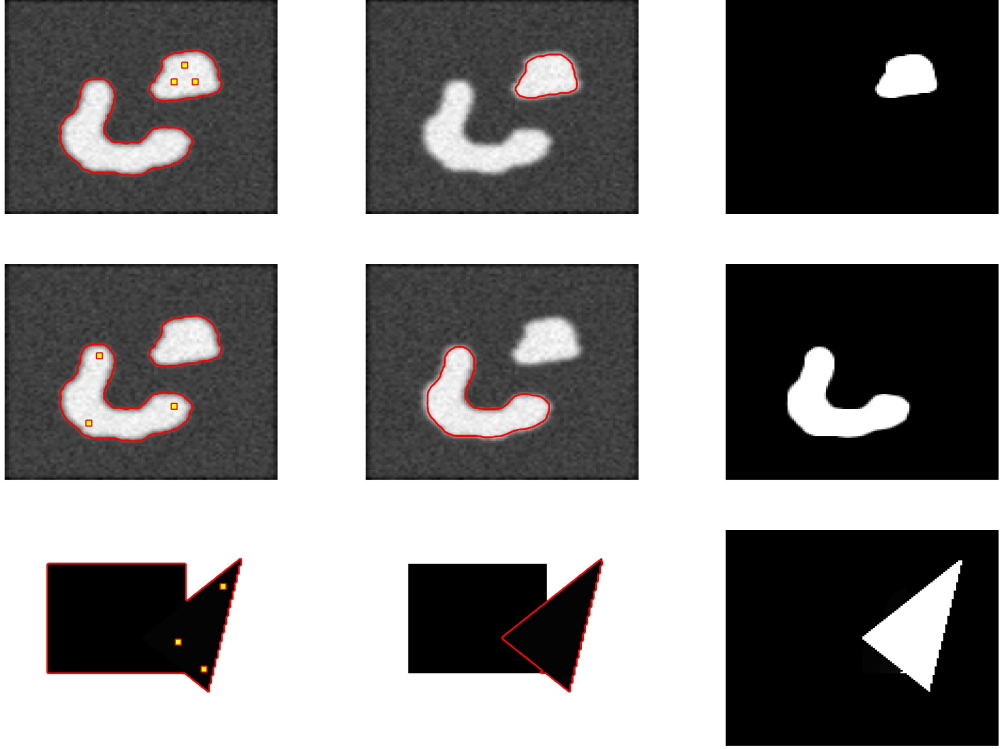


Figure 5.3: Test Set 2. Results for CDSS, for three test problems (given by rows 1-3). From left to right: initialisation (with user input set \mathcal{A}), final contour, object selected.

5.5.3 Test Set 3 (Improved AOS Method)

In these tests we recall the improved AOS methods introduced in Chapter 4. In Fig. 5.6 we present results for AOS1 with a time step of $\tau = 0.01$, and varying the parameter ς in the AOS interval I_ς . This demonstrates that for larger time steps, varying the width of the interval I_ς can improve the convergence of the scheme, such that a stopping criterion can be applied. One notes here that the contour (Γ^*) shown is accurate for both, due to the thresholding procedure, but the segmentation function u^* is closer to binary and stable as the iterations progress for $\varsigma = 0.1$.

In Fig. 5.7 we present results for AOS2 with a time step of $\tau = 1$, and varying the parameter ς . This demonstrates that varying the width of the interval I_ς can improve convergence, such that a stopping criterion can be applied. For the smaller interval, with $\varsigma = 0.1$, the segmentation function u^* is noisy such that the thresholding procedure does not produce an accurate result for Γ^* . Here, the selection of $\tau = 1$ is almost arbitrary, and we see similar results for larger time steps.

These results underline the findings in Chapter 4, where the improved AOS schemes, AOS1 and AOS2, can improve the convergence behaviour of u^* such that a reasonable stopping criterion can be applied and a final contour Γ^* can be computed in good time. Here, we demonstrate their proficiency in a setting where the fitting term is unknown. The spikes in the residuals in Figs. 5.6 and 5.7 correspond with updates of the intensity constants, c_1 and c_2 . It is important to note that their initial values are close to their

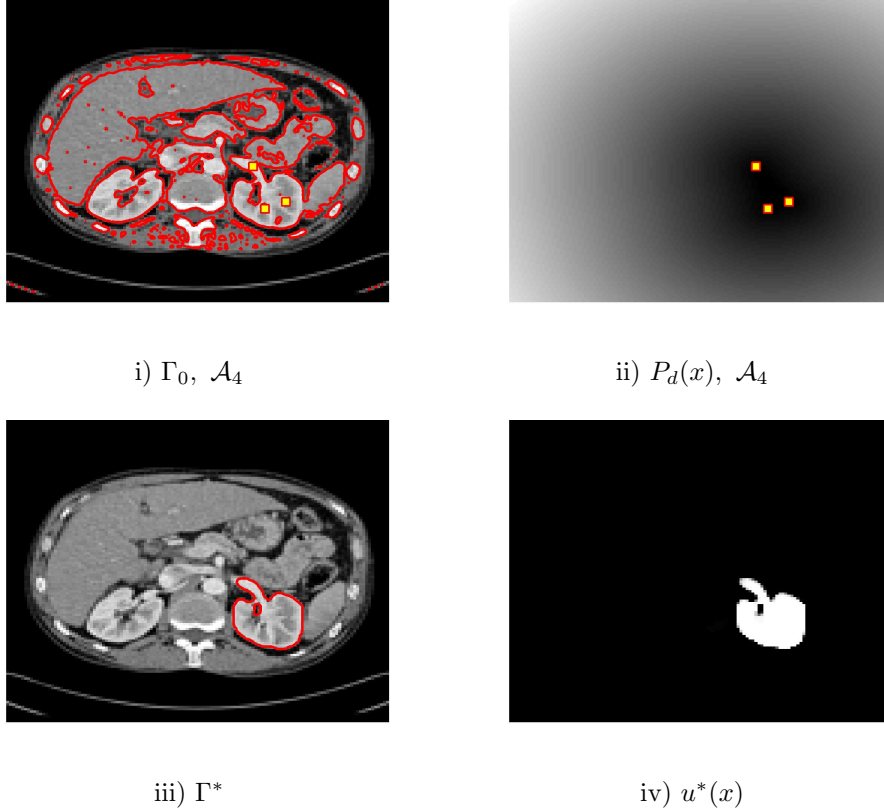


Figure 5.4: Test Set 2. User input set 1 for CDSS. From left to right, top to bottom: initialisation, P_d function (with user input set \mathcal{A}_4), final contour, object selected.

converged values, due to the user input, and thus the problem is not completely convex. This can be considered in the future following work such as Brown et al. [20].

5.5.4 Test Set 4 (Medical Applications)

Finally, we present results demonstrating the ability of the model to accurately partition a foreground object of similar intensity to other objects in the image. In Figs. 5.8 and 5.9 we present results for two medical images. Based on minimal user input (three or four markers), objects can be selected for challenging examples with potential applications. We can see from ii) and iv) that the converged results for $u^*(x)$ are consistent with the ideas introduced in Chapter 4, in the sense that the result is close to binary. This is vital to the accuracy and reliability of the thresholding procedure, such that Γ^* is very close to the true boundary of the object. We can also observe the edge function, $g(x)$, in the weighted TV term, and the distance selection term, $P_d(x)$, based on the user input set.

5.6 Remarks

In this chapter we discussed the drawbacks of current selective segmentation models and proposed a new model where a global minimiser can be found independently of initialisation. One notes here that, like with the global case of Chan, Esedoglu, and

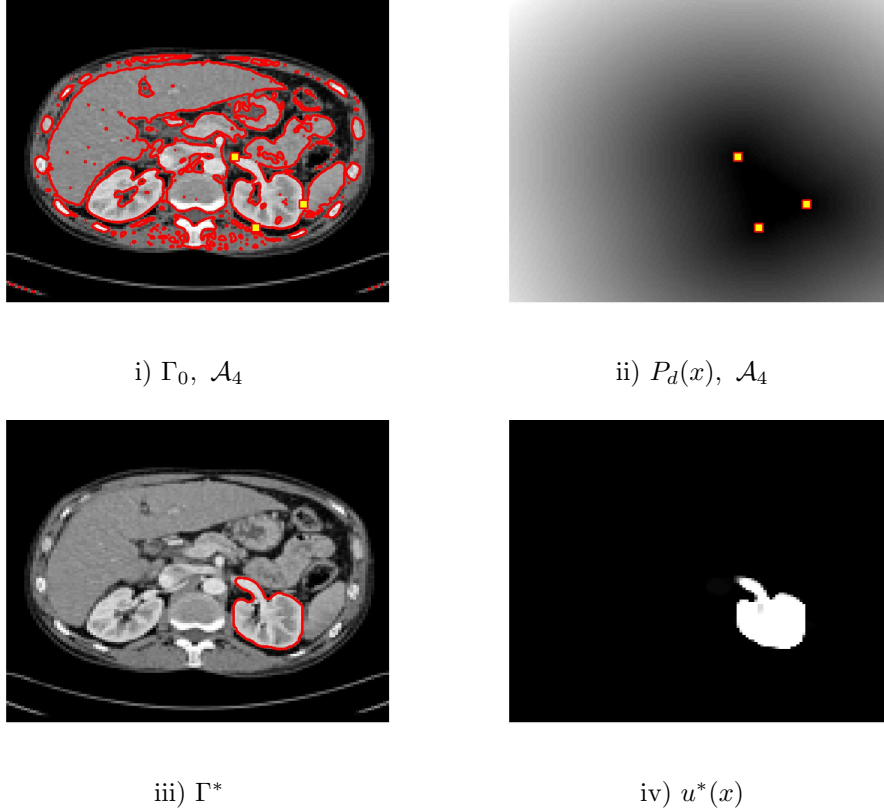
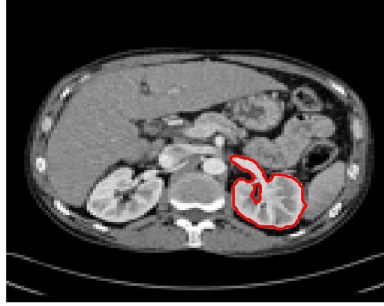


Figure 5.5: Test Set 2. User input set 2 for CDSS. From left to right, top to bottom: initialisation, P_d function (with user input set \mathcal{A}_5), final contour, object selected.

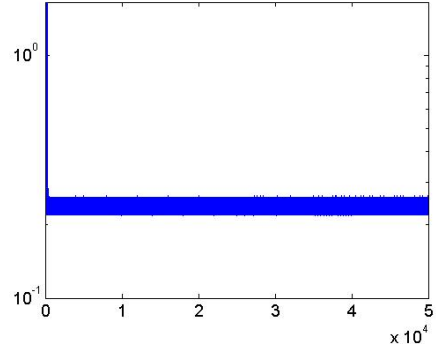
Nikolova [30], in theory this relies on the fitting function being fixed. In other words, the intensity constants must be known. However, in practice superior results can be achieved if at least one of c_1 or c_2 is minimised. Given that the user input set provides a reasonable approximation of the intensity of the target object it is natural to expect this to be reasonably close to the true value, and thus local minima for these variables is unlikely. We refer the reader to Brown et al. [20] for a completely convex formulation, and related considerations.

The work of [30] and [105] motivated a proposed nonconvex selective model, and we detail its convex reformulation. In the nonconvex case, our model performs well in comparison to Rada-Chen [105], and we demonstrate that global minimisers of the corresponding nonconvex formulation can be found. Vital to the success of our proposed model, is its lack of dependence on the user input. In potential applications of selective segmentation models an over-reliance on comprehensive and specific user input is prohibitive. In our results we show that simple user input (i.e. three or four markers) can produce a good result in a difficult example and, crucially, a similar result can be obtained for a very different marker set.

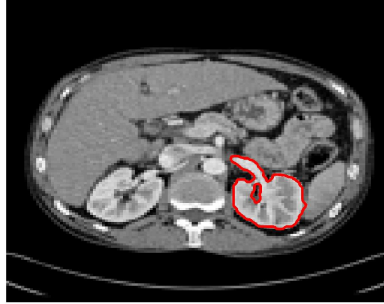
We also provide results which support the conclusions about the improved AOS method introduced in chapter 4, and demonstrate its applicability in practice. The additional results we provide demonstrate the potential applications of selective segmentation. Here, no knowledge of the object is known a priori and yet we can achieve sat-



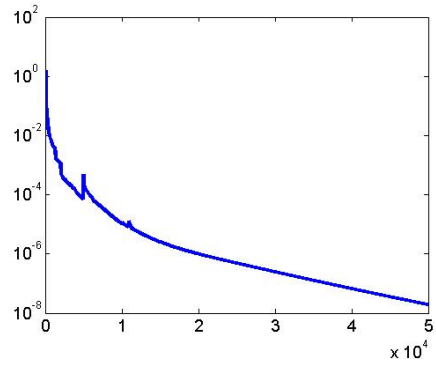
i) Γ^* , $\varsigma = 0.01$



ii) R , $\varsigma = 0.01$



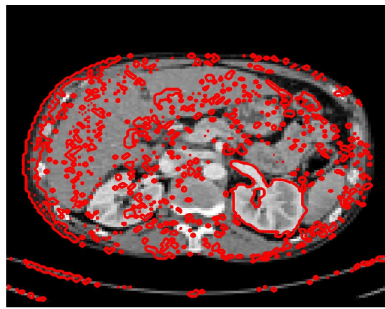
iii) Γ^* , $\varsigma = 0.1$



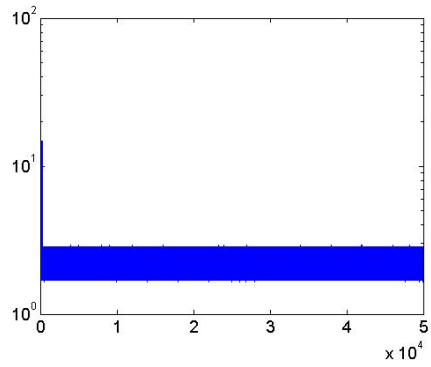
iv) R , $\varsigma = 0.1$

Figure 5.6: Test Set 3. Results for $AOS_1, \tau = 10^{-2}$ for CDSS. Row 1 is for $\varsigma = 0.01$, row 2 is for $\varsigma = 0.1$. From left to right: final contour and residual for u (with number of iterations).

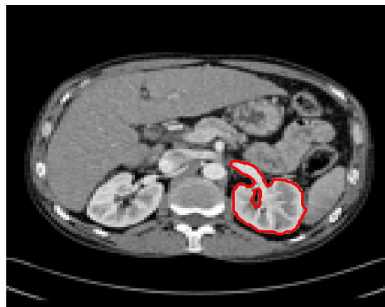
isfactory results for difficult examples from medical imaging using a piecewise-constant assumption. Later in the thesis, we consider the case of intensity inhomogeneity and the challenge of incorporating prior knowledge in the model. This expands the scope of what can be achieved with image segmentation methods.



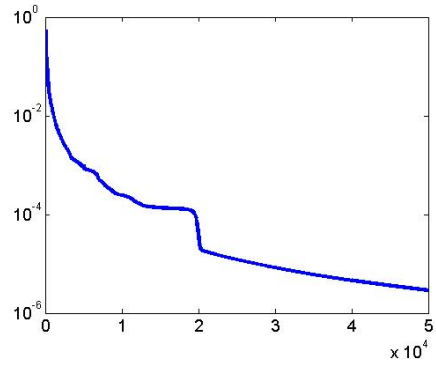
i) Γ^* , $\varsigma = 0.1$



ii) R , $\varsigma = 0.1$

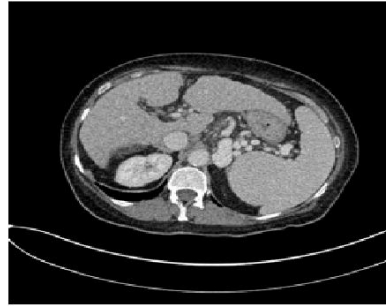


iii) Γ^* , $\varsigma = 0.5$

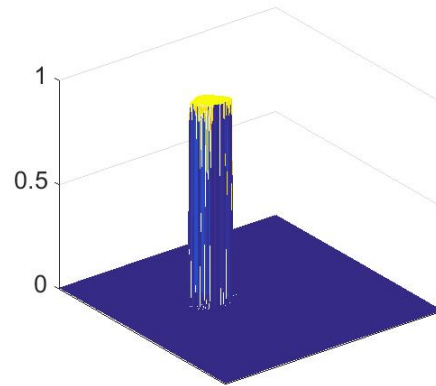


iv) R , $\varsigma = 0.5$

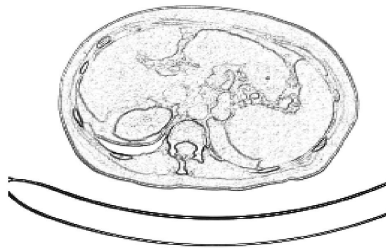
Figure 5.7: Test Set 3. Results for $AOS_2, \tau = 1$ for CDSS. Row 1 is for $\varsigma = 0.1$, row 2 is for $\varsigma = 0.5$. From left to right: final contour and residual for u (with number of iterations).



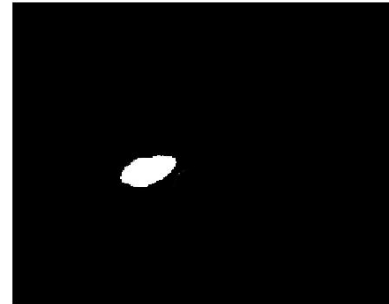
i) $z(x)$



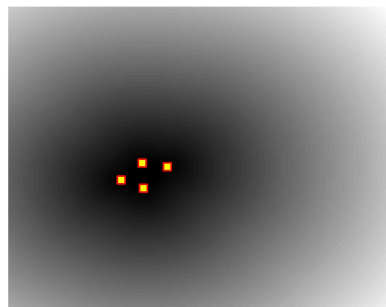
ii) $u^*(x)$



iii) $g(x)$



iv) $u^*(x)$

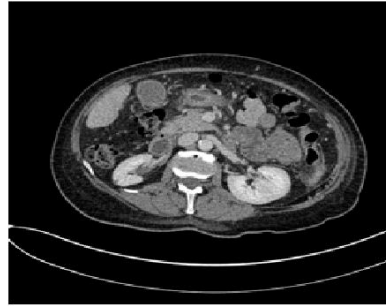


v) $P_d(x)$

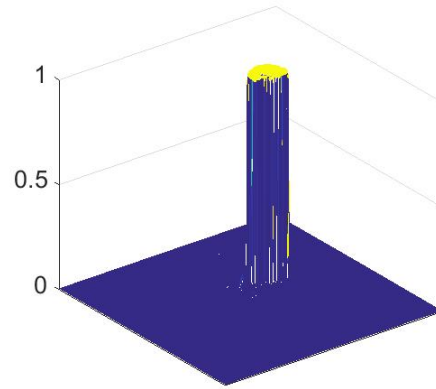


vi) Γ^*

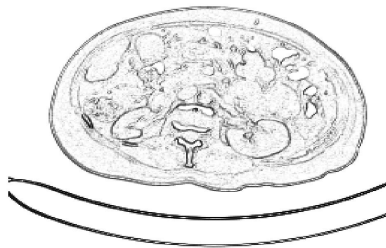
Figure 5.8: Test Set 4, Image 1. Results for CDSS. i) Observed Image, $z(x)$. ii) The converged segmentation function, $u^*(x)$. iii) The edge detection function, $g(x)$, for the weighted TV term. iv) An alternative view of the segmentation function, $u^*(x)$. v) The distance selection term, $P_d(x)$, based on the user input set \mathcal{A} . vi) The thresholded contour, Γ^* , demonstrating an accurate result for the object of interest.



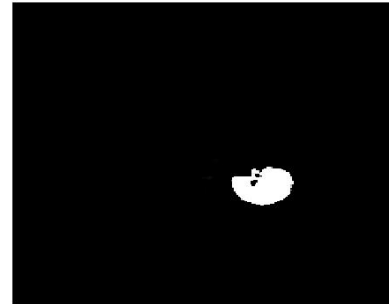
i) $z(x)$



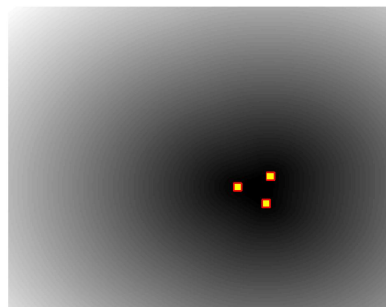
ii) $u^*(x)$



iii) $g(x)$



iv) $u^*(x)$



v) $P_d(x)$



vi) Γ^*

Figure 5.9: Test Set 4, Image 2. Results for CDSS. i) Observed Image, $z(x)$. ii) The converged segmentation function, $u^*(x)$. iii) The edge detection function, $g(x)$, for the weighted TV term. iv) An alternative view of the segmentation function, $u^*(x)$. v) The distance selection term, $P_d(x)$, based on the user input set \mathcal{A} . vi) The thresholded contour, Γ^* , demonstrating an accurate result for the object of interest.

Chapter 6

Segmentation with Intensity Inhomogeneity

6.1 Introduction

Selecting objects in an image based on intensity similarity has been widely studied over the last twenty years [30, 33, 89] and is particularly challenging in cases of intensity inhomogeneity, which we will address in this chapter. In Chapter 3 we discuss the two-phase piecewise-constant case of the Mumford-Shah formulation, which is most suitable for images with simple and homogeneous features where the intensity variation is limited. However, it has been applied to many different types of synthetic and real images after some adjustments to the formulation. Recent work has incorporated bias field estimation to allow for intensity inhomogeneity, with great success in terms of segmentation quality. However, the framework and assumptions involved lead to inconsistencies in the method that can adversely affect results. This chapter focuses on our approach to segmentation with intensity inhomogeneity, where we introduce an additional constraint to the bias field framework in the context of the convex relaxation method.

We first introduce the concept of the bias field framework for two-phase segmentation. In the following we discuss generalising the piecewise-constant formulation, to approximate minimisers of the original Mumford-Shah formulation. We now recall some familiar definitions from previous chapters, to make the introduction of our method clearer. There are also differences in notation here with earlier chapters, and wider conventional usage. For example, in the Mumford-Shah functional [89] we replace u with w in order to avoid confusion between the piecewise-smooth approximation of the image and the segmentation function in globally convex segmentation (GCS) referred to in previous chapters. Given an image $z(x)$ in a bounded domain $\Omega \subset \mathbb{R}^2$, we look for an edge Γ that partitions Ω into regions $\{\Omega_i, i = 1, 2, \dots, N\}$ in $\Omega \setminus \Gamma$. We recall the Mumford-Shah functional [89], first introduced in Chapter 3:

$$E^{MS}(\Gamma, w) = |\Gamma| + \lambda \int_{\Omega} (z - w)^2 dx + \mu \int_{\Omega \setminus \Gamma} |\nabla w| dx, \quad (6.1)$$

where $\mu, \lambda > 0$ are weighting parameters, and $|\Gamma|$ denotes the length of the edge curve

Γ , the boundary between regions Ω_i . Mumford and Shah [89] demonstrated that, theoretically, the existence and regularity of minimisers of this functional can be achieved, and Tsai et al. [122] and Vese and Chan [123] used the variational level set method of Osher et al. [95] and Zhao et al. [143] to minimise (6.1). Chan and Vese [33] considered a functional that was a particular case of (6.1), the two-phase piecewise-constant example, i.e. $|\nabla w| = 0$ in each region, and $N = 2$ in (6.1).

$$E^{CV}(\Gamma, c_1, c_2) = |\Gamma| + \lambda \left(\int_{\Omega_1} (z - c_1)^2 dx + \int_{\Omega_2} (z - c_2)^2 dx \right). \quad (6.2)$$

The Chan-Vese framework has been generalised by the introduction of new fitting terms to incorporate extensive intensity inhomogeneity, such as Li et al. [82, 81] who introduced a region scalable fitting energy and local cluster method. Jung et al. [70] introduced a nonlocal active contour model utilising distance functions. Brox and Cremers [21] and Lankton and Tannenbaum [77] introduced new local models, incorporating Gaussian kernel functions. Recent work related to this area includes the work of Ali et al. [3], who form fitting terms using multiplicative and difference image data, and L_0 regularisation for simultaneous bias correction and segmentation by Duan et al. [49].

As discussed in previous chapters the drawback of the Chan-Vese approach is its lack of convexity, and there are advantages of considering GCS instead. A recent model that combines this convex relaxation framework and segmentation with intensity inhomogeneity is D. Chen et al. [37], and our work focuses on aiming to improve their formulation in order to achieve more robust results in difficult examples. The authors [37] assume the 'true' image data is formulated [2, 84] as

$$T = \sum_i c_i \chi_i, \quad i = 1, 2, \dots, N, \quad (6.3)$$

where c_i are intensity constants, and χ_i are characteristic functions of the regions Ω_i . It is based on the idea that the image can be modelled as

$$z(x) = B(x)T + \eta, \quad x \in \Omega_i, \quad i = 1, 2, \dots, N, \quad (6.4)$$

where η is additive noise. Here, as with the rest of the work in this thesis, we consider the two-phase case, i.e. $N = 2$. D. Chen et al. [37] aim to estimate the bias field B and recover the 'true' image T . However, a lack of convergence of the bias field and the intensity constants, means that recovering an accurate T is not possible. We propose introducing a constraint on these variables to correct this inconsistency.

This chapter is organised as follows. In Section 6.2 we detail the Variant Mumford-Shah Model [37], briefly introduced above, and discuss the problems with recovering the 'true' image, in particular the lack of convergence of c_1 and c_2 due to the formulation. In Section 6.3 we detail the introduction of a constraint to the work of D. Chen et al. [37] in order to automatically establish feasible intensity constants, and ensure the convergence of all variables being minimised. We discuss how this alters the minimisation of the bias field, how the functional is iteratively minimised, and details of the numerical implementation. We also highlight the link the proposed method

provides between Mumford-Shah [89] and Chan-Vese [33]. In Section 6.3.4 we include experimental results that measure the accuracy of the proposed method compared to Variant Mumford-Shah, and demonstrate the convergence of the intensity constants for examples used in [37]. We extend this idea to selective segmentation in Section 6.4 by incorporating the distance selection term introduced in Chapter 5, and include experimental results for one challenging case. We discuss the benefits of the proposed method, and make some concluding remarks in Section 6.5.

6.2 Variant Mumford-Shah Model

The Variant Mumford-Shah Model (VMS) by D. Chen et al. [37] is formulated as follows:

$$E^{VMS}(\Gamma, c_1, c_2, B) = |\Gamma| + \lambda \int_{\Omega} ((z - Bc_1)^2 \chi_1 + (z - Bc_2)^2 \chi_2) dx + \mu \int_{\Omega} |\nabla B|^2 dx, \quad (6.5)$$

where λ and μ are weighting parameters. The idea is that the intensity constants represent the 'true' image, and the bias field B varies such that their combination gives a piecewise-smooth approximation of z (with respect to the Mumford-Shah [89] formulation (6.1)), given by

$$w^{VMS} = Bc_1\chi_1 + Bc_2\chi_2. \quad (6.6)$$

The functional (6.5) is minimised iteratively by the following steps. Step (1): For fixed characteristic functions χ_1 and χ_2 , and intensity constants c_1 and c_2 , minimise (6.5) with respect to bias field estimator B . Based on the work of Nielsen et al. [93] and Brox and Cremers [21], the exact minimiser can be well approximated. Step (2): For fixed characteristic functions χ_1 and χ_2 , and bias field estimator B , minimise (6.5) with respect to intensity constants c_1 and c_2 . These can be computed precisely. Step (3): For fixed intensity constants c_1 and c_2 , and bias field estimator B , minimise (6.5) with respect to χ_1, χ_2 . Based on the work of Chan, Esedoglu, and Nikolova [30] minimising two characteristic functions can be achieved by with the familiar convex relaxation methods discussed in detail in Chapters 3 and 4.

6.2.1 Convergence Behaviour of VMS

In Fig. 6.1, we demonstrate a result for VMS that is also used in [37], and is of comparable quality. However, the question remains: based on the image model described above (6.3), what is the 'true' image? Whilst the joint minimisation of (6.5) with respect to c_1, c_2, B and Γ is nonconvex, and therefore we cannot determine the correct c_1 and c_2 precisely, there is a problem with the current framework, which we will now discuss. In Fig. 6.1 (after 1000 iterations), we show that the values of the intensity constants continually rise, such that $c_1 = 9.1 \times 10^4$ and $c_2 = 6.4 \times 10^4$. The convergence of w^{VMS} (6.6) comes from the reduction in scale of B . To demonstrate this, after the same number of iterations $\|B\|_F = 4.7 \times 10^{-4}$ (where $\|\cdot\|_F$ is the Frobenius norm). This

motivates our proposal for modifying VMS, in the form of an additional constraint, that can automatically control the scale of c_1, c_2 and B .

To explain this phenomenon let's examine the VMS functional (6.5). The smoothness penalty included, which we denote E^B , is similar to the penalty enforced in the Mumford-Shah functional (6.1), except that it applies throughout the domain. We denote the fitting energy E^F , and it is again similar to the Mumford-Shah fitting energy:

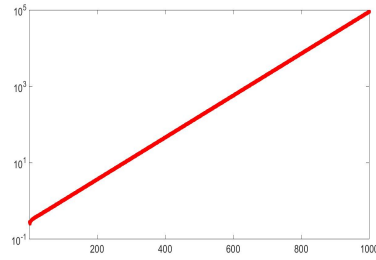
$$E^B = \int_{\Omega} |\nabla B|^2 dx, \quad E^F = \int_{\Omega} ((z - Bc_1)^2 \chi_1 + (z - Bc_2)^2 \chi_2) dx. \quad (6.7)$$

However, crucially, the Mumford-Shah fitting term only involves one variable, w . The VMS fitting term involves the products Bc_1 and Bc_2 . This means that a change in one variable doesn't necessarily alter the energy, as long as the other variable changes accordingly. In practice that means that the minimum of the VMS functional is attained when $E^B \rightarrow 0$, despite the convergence of E^F . This is due to the lack of convergence of the intensity constants c_1 and c_2 , but this contradicts the assumptions of recovering the 'true' image (6.3) discussed in Section 6.1. This is demonstrated in Fig. 6.1.

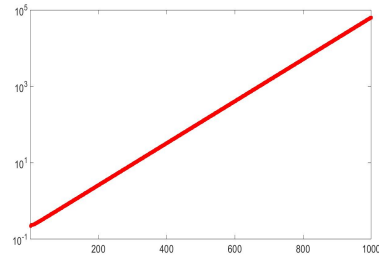
6.3 Stabilised Bias Field

VMS produces a piecewise-smooth approximation of the image (in the Mumford-Shah sense [89]), given by w^{VMS} (6.6). However, it does not give values for c_1 , and c_2 that are consistent with the observed image. It is possible to manually rescale these without changing w^{VMS} , but this is not a sensible approach as these values are unknown by definition. The immediate question is: is it possible to incorporate constraints into the formulation in a reliable way, i.e. can we use information in the image to automatically restrict the scale of B, c_1 and c_2 ? There are two obvious approaches. The first is to constrain the values of c_1 or c_2 . The situation when the optimal intensity constants are not known a priori has been studied by Brown et al. [20] in the piecewise-constant case, but not in cases of intensity inhomogeneity. It is worth considering how this method could be incorporated in the presence of a bias field function, however we do not discuss this here. The second is to control the scale of the bias field, B . We therefore consider how to introduce a constraint in such a way that it provides a link between the piecewise-constant and piecewise-smooth approximations of z that are consistent with the image, which we will return to later.

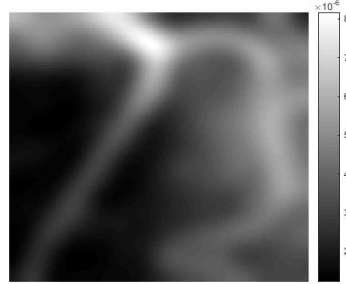
With VMS, B is encouraged to be close to 0, which leads to the lack of convergence for c_1 and c_2 . To prevent this we propose a new model we call Stabilised Bias Field (SBF), with the introduction of an additional constraint that encourages B to be close to a positive constant. However, this alters the minimisation step for the bias field from [37]. We now consider how to obtain this with the addition of this constraint. To distinguish between the two methods we refer to the bias field in SBF as \tilde{B} . The new



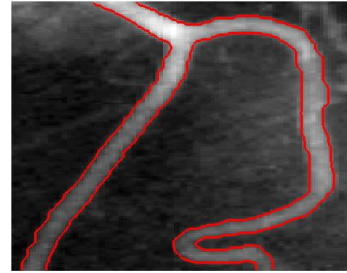
(a) c_1 Progression



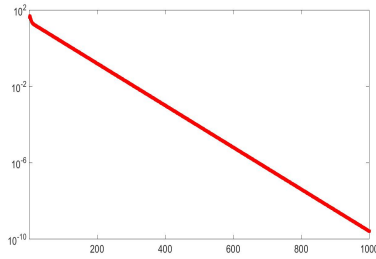
(b) c_2 Progression



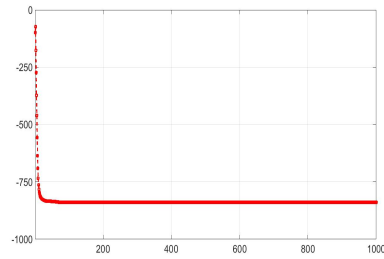
(c) Bias Field, $B(x)$



(d) $z(x) \in [0, 1], \Gamma^*$



(e) E^B Progression



(f) E^F Progression

Figure 6.1: VMS Results. Convergence Behaviour. The first row shows the lack of convergence for the intensity constants, giving $c_1 = 9.1 \times 10^4$ and $c_2 = 6.4 \times 10^4$ after 1000 iterations. The second row shows the scale of the bias field, $B(x)$, on the left and the image, $z(x)$, and computed contour, Γ^* , on the right. The bottom row shows the progression of the energies E^B and E^F (6.7) (iterations on horizontal axis).

formulation is given as follows:

$$\begin{aligned}
E^{SBF}(\Gamma, c_1, c_2, \tilde{B}) &= |\Gamma| + \lambda \int_{\Omega} \left((z - \tilde{B}c_1)^2 \chi_1 + (z - \tilde{B}c_2)^2 \chi_2 \right) dx \\
&\quad + \mu \int_{\Omega} |\nabla \tilde{B}|^2 dx + \gamma \int_{\Omega} (\tilde{B} - \mathbf{s})^2 dx,
\end{aligned} \tag{6.8}$$

where \mathbf{s}, γ are positive parameters. We intend to use the framework of VMS to approximate the exact minimiser of (6.8) for \tilde{B} . With this in mind, the previous formulation is equivalent to

$$\begin{aligned}
E^{SBF}(\Gamma, c_1, c_2, \tilde{B}) &= \lambda \int_{\Omega} \left[(z - \tilde{B}c_1)^2 + \frac{\gamma}{\lambda} (B - \mathbf{s})^2 \right] \chi_1 dx \\
&\quad + \lambda \int_{\Omega} \left[(z - \tilde{B}c_2)^2 + \frac{\gamma}{\lambda} (\tilde{B} - \mathbf{s})^2 \right] \chi_2 dx \\
&\quad + |\Gamma| + \mu \int_{\Omega} |\nabla \tilde{B}|^2 dx,
\end{aligned}$$

where the new constraint has been incorporated into the fitting term. We can reformulate this as follows, first looking at the χ_1 term:

$$\begin{aligned}
\left[(z - \tilde{B}c_1)^2 + \frac{\gamma}{\lambda} (\tilde{B} - \mathbf{s})^2 \right] &= \tilde{B}^2 (c_1^2 + \tilde{\gamma}) - 2\tilde{B} (c_1 z + \tilde{\gamma} \mathbf{s}) + (z^2 + \tilde{\gamma} \mathbf{s}^2) \\
&= (c_1^2 + \tilde{\gamma}) \left[\tilde{B} - \frac{c_1 z + \tilde{\gamma} \mathbf{s}}{c_1^2 + \tilde{\gamma}} \right]^2 + f_1(z, c_1, \mathbf{s}, \tilde{\gamma}) \\
&= \left[\frac{c_1 z + \tilde{\gamma} \mathbf{s}}{\sqrt{c_1^2 + \tilde{\gamma}}} - \tilde{B} \sqrt{c_1^2 + \tilde{\gamma}} \right]^2 + f_1(z, c_1, \mathbf{s}, \tilde{\gamma}),
\end{aligned}$$

where $\tilde{\gamma} = \frac{\gamma}{\lambda}$ and $f_1(z, c_1, \mathbf{s}, \tilde{\gamma}) = \frac{z^2 + \tilde{\gamma} \mathbf{s}^2}{c_1^2 + \tilde{\gamma}}$. In a similar way, for the χ_2 term:

$$\left[(z - \tilde{B}c_2)^2 + \frac{\gamma}{\lambda} (\tilde{B} - \mathbf{s})^2 \right] = \left[\frac{c_2 z + \tilde{\gamma} \mathbf{s}}{\sqrt{c_2^2 + \tilde{\gamma}}} - \tilde{B} \sqrt{c_2^2 + \tilde{\gamma}} \right]^2 + f_2(z, c_2, \mathbf{s}, \tilde{\gamma}),$$

where $f_2(z, c_2, \mathbf{s}, \tilde{\gamma}) = \frac{z^2 + \tilde{\gamma} \mathbf{s}^2}{c_2^2 + \tilde{\gamma}}$. Therefore (6.8) is equivalent to

$$\begin{aligned}
E^{SBF}(\Gamma, c_1, c_2, \tilde{B}) &= \lambda \int_{\Omega} \left[\frac{c_1 z + \tilde{\gamma} \mathbf{s}}{\sqrt{c_1^2 + \tilde{\gamma}}} - \tilde{B} \sqrt{c_1^2 + \tilde{\gamma}} \right]^2 \chi_1 dx \\
&\quad + \lambda \int_{\Omega} \left[\frac{c_2 z + \tilde{\gamma} \mathbf{s}}{\sqrt{c_2^2 + \tilde{\gamma}}} - \tilde{B} \sqrt{c_2^2 + \tilde{\gamma}} \right]^2 \chi_2 dx \\
&\quad + \int_{\Omega} f_1(z, c_1, \mathbf{s}, \tilde{\gamma}) \chi_1 dx + \int_{\Omega} f_2(z, c_2, \mathbf{s}, \tilde{\gamma}) \chi_2 dx \\
&\quad + |\Gamma| + \mu \int_{\Omega} |\nabla \tilde{B}|^2 dx.
\end{aligned}$$

Minimising $E^{SBF}(\Gamma, c_1, c_2, \tilde{B})$ with respect to \tilde{B} is given by

$$\min_{\tilde{B}} \left\{ \lambda \int_{\Omega} \left(\left[z_1 - \tilde{B} \tilde{c}_1 \right]^2 \chi_1 + \left[z_2 - \tilde{B} \tilde{c}_2 \right]^2 \chi_2 \right) dx + \mu \int_{\Omega} |\nabla \tilde{B}|^2 dx \right\}, \tag{6.9}$$

since $f_1(z, c_1, \mathbf{s}, \tilde{\gamma}), f_2(z, c_2, \mathbf{s}, \tilde{\gamma})$, and $|\Gamma|$ are not dependent on \tilde{B} . Here

$$z_1 = \frac{c_1 z + \tilde{\gamma} \mathbf{s}}{\sqrt{c_1^2 + \tilde{\gamma}}}, \quad \tilde{c}_1 = \sqrt{c_1^2 + \tilde{\gamma}}, \quad z_2 = \frac{c_2 z + \tilde{\gamma} \mathbf{s}}{\sqrt{c_2^2 + \tilde{\gamma}}}, \quad \text{and} \quad \tilde{c}_2 = \sqrt{c_2^2 + \tilde{\gamma}}. \quad (6.10)$$

In the same way as VMS [37], we can approximate the exact minimiser of (6.9) with a Gaussian G_σ :

$$\tilde{B} = \frac{\tilde{c}_1 z_1 \chi_1 + \tilde{c}_2 z_2 \chi_2}{\tilde{c}_1^2 \chi_1 + \tilde{c}_2^2 \chi_2} * G_\sigma. \quad (6.11)$$

6.3.1 Relationship to Chan-Vese and Mumford-Shah

We now discuss how the proposed model relates to the two important works discussed earlier in Section 6.1, and Chapter 3. The SBF functional is given as

$$\begin{aligned} E^{SBF}(\Gamma, c_1, c_2, \tilde{B}) &= |\Gamma| + \lambda \int_{\Omega} \left((z - \tilde{B}c_1)^2 \chi_1 + (z - \tilde{B}c_2)^2 \chi_2 \right) dx \\ &\quad + \mu \int_{\Omega} |\nabla \tilde{B}|^2 dx + \gamma \int_{\Omega} (\tilde{B} - \mathbf{s})^2 dx. \end{aligned} \quad (6.12)$$

It relates to Mumford-Shah in the same sense that VMS does. That is, we can compute a piecewise-smooth approximation of the image,

$$w^{SBF} = \tilde{B}c_1 \chi_1 + \tilde{B}c_2 \chi_2, \quad (6.13)$$

except that the values computed correspond to the observed image $z \in [0, 1]$, and the variables converge reliably. However, it also relates to the Chan-Vese functional. If $\mathbf{s} = 1$, and $\gamma \rightarrow \infty$ we have the Chan-Vese formulation (6.2):

$$E^{CV}(\Gamma, c_1, c_2) = |\Gamma| + \lambda \int_{\Omega} \left((z - c_1)^2 \chi_1 + (z - c_2)^2 \chi_2 \right) dx.$$

6.3.2 Iterative Minimisation of SBF Formulation

We now detail how to minimise the functional (6.8), in line with the method of D. Chen et al. [37], in order to effectively compare our proposed method against VMS. The SBF Model is given as follows:

$$\min_{\Gamma, c_1, c_2, \tilde{B}} \left\{ E^{SBF}(\Gamma, c_1, c_2, \tilde{B}) = |\Gamma| + \lambda \int_{\Omega} \left((z - \tilde{B}c_1)^2 \chi_1 + (z - \tilde{B}c_2)^2 \chi_2 \right) dx + \mu \int_{\Omega} |\nabla \tilde{B}|^2 dx + \gamma \int_{\Omega} (\tilde{B} - \mathbf{s})^2 dx \right\}.$$

This is minimised iteratively (e.g. the iterative process method, Li et al. [81]) by the following steps:

- (1) For fixed characteristic functions χ_1 and χ_2 , and intensity constants c_1 and c_2 , minimise (6.12) with respect to bias field estimator \tilde{B} .
- (2) For fixed characteristic functions χ_1 and χ_2 , and bias field estimator \tilde{B} , minimise (6.12) with respect to intensity constants c_1 and c_2 .

- (3) For fixed intensity constants c_1 and c_2 , and bias field estimator \tilde{B} , minimise (6.12) with respect to characteristic functions χ_1 and χ_2 .

We provide a summary of how each step is minimised in the following:

Step (1): detailed in the previous section. It can be approximated, according to the work of Nielsen et al. [93] and Brox and Cremers [21] and as discussed by D. Chen et al. [37], by:

$$\tilde{B} = \frac{\tilde{c}_1 z_1 \chi_1 + \tilde{c}_2 z_2 \chi_2}{\tilde{c}_1^2 \chi_1 + \tilde{c}_2^2 \chi_2} * G_\sigma.$$

Step (2): minimising with respect to c_1 and c_2 gives

$$c_1 = \frac{\int_{\Omega} z(x) \tilde{B}(x) \chi_1 dx}{\int_{\Omega} \tilde{B}^2(x) \chi_1 dx}, \quad c_2 = \frac{\int_{\Omega} z(x) \tilde{B}(x) \chi_2 dx}{\int_{\Omega} \tilde{B}^2(x) \chi_2 dx}. \quad (6.14)$$

Step (3): achieved by the following minimisation:

$$\min_{\chi_1, \chi_2} \left\{ |\Gamma| + \lambda \int_{\Omega} \left((z - \tilde{B}c_1)^2 \chi_1 + (z - \tilde{B}c_2)^2 \chi_2 \right) dx \right\} \quad (6.15)$$

Minimising two characteristic functions can be achieved with convex relaxation methods, such as the method introduced in Chapter 4:

$$\min_{0 \leq u \leq 1} \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} \left((z - \tilde{B}c_1)^2 - (z - \tilde{B}c_2)^2 \right) u(x) dx \right\}, \quad (6.16)$$

In [37] the authors use the dual formulation of Chambolle [23, 18], as detailed in previous chapters. We use the dual formulation here in order to fairly compare between VMS and SBF.

6.3.3 Numerical Implementation

We now provide details of implementing the three steps above. We follow the work of D. Chen et al. [37], who use slight variations on the formulation, in order to be consistent with VMS. The intensity constants are computed using smooth region descriptors $H_\varsigma^{(1)}$ and $H_\varsigma^{(2)} = 1 - H_\varsigma^{(1)}$ instead of characteristic functions χ_1 and χ_2 respectively. This descriptor is defined as follows:

$$H_\varsigma^{(1)}(\varphi(x)) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan(\varphi * G_\varsigma) \right), \quad x \in \Omega.$$

The variable $\varphi(x)$ is given by

$$\varphi(x) = \begin{cases} c, & \text{for } x \in \Omega : u(x) > \epsilon, \\ -c, & \text{for } x \in \Omega : u(x) \leq \epsilon, \end{cases}$$

where $\epsilon \in [0, 1]$, $c = 2$ and $\varsigma = 1$. This adjusts the computation of the intensity constants (6.14) to

$$c_1 = \frac{\int_{\Omega} z(x) \tilde{B}(x) H_{\zeta}^{(1)} dx}{\int_{\Omega} \tilde{B}^2(x) H_{\zeta}^{(1)} dx}, \quad c_2 = \frac{\int_{\Omega} z(x) \tilde{B}(x) H_{\zeta}^{(2)} dx}{\int_{\Omega} \tilde{B}^2(x) H_{\zeta}^{(2)} dx}. \quad (6.17)$$

The Gaussian kernel, G_{σ} , is truncated as a $\varrho \times \varrho$ mask, where ϱ is the smallest odd number greater than $4\sigma + 1$ (σ is the standard deviation of the Gaussian kernel [81]). Other parameters mentioned in Section 6.3.2 are set as follows: $\tau = 1/8, \theta = 1/3, \epsilon = 1/2$.

The primary motivation of this model is to have convergence of the intensity constants c_1 and c_2 . The value of \mathbf{s} determines the size of these values; as $\mathbf{s} \rightarrow 0$, $c_1, c_2 \rightarrow \infty$, as discussed in Section 6.2.1. For consistency, it is desirable that $c_1, c_2 \in [0, 1]$ given $z \in [0, 1]$. With that in mind, a natural selection is $\mathbf{s} = 1$ given that the intensity constants are then related to the average value of $z(x)$ inside and outside the contour. Additionally, SBF is then clearly related to Chan-Vese [33] as detailed in Section 6.3.1. It can be seen from (6.10) that the update for the bias field, \tilde{B} (given by (6.11)), is dependent on the parameters σ, \mathbf{s} , and γ , whereas the updates for c_1 and c_2 (given by (6.17)) are not directly dependent on these parameters. In order to complete Step (3), given by (6.15) and dependent on λ , we follow Chambolle's dual formulation as introduced by Bresson et al. [18] and Chambolle [23], and discussed in Section 3.3.1. In Chapter 4 we conclude that an appropriate stopping criterion, δ , can be chosen as 0.1. This is discussed in detail in Section 4.5.2, and supported by Figs. 4.10 and 4.11. We set the maximum number of iterations, denoted by *maxit*, at 1000. It is worth noting that in Figs. 6.3, 6.5, and 6.7 we present the convergence behaviour for c_1 and c_2 where the stopping criterion has been removed so we can observe this aspect of the results in full. However, in general we use δ in a similar way to Chapter 4 as presented in Algorithm 2.

Algorithm 2 Stabilised Bias Field: $\Omega_1^* \leftarrow SBF(z, \text{maxit}, \delta, \lambda, \sigma, \mathbf{s}, \gamma)$

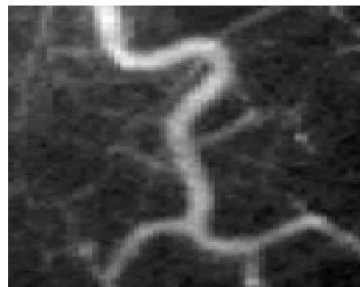
- 1: Initialise $u^{(0)}$, estimate $c_1^{(1)}, c_2^{(1)}, \tilde{B}^{(1)}$.
 - 2: **for** $\ell \leftarrow 1 : \text{maxit}$ **do**
 - 3: **if** $\text{mod}(\ell, 2) = 0$ **then**
 - 4: Calculate $\tilde{B}^{(\ell)}$ (using (6.11)), dependent on σ, \mathbf{s} and γ .
 - 5: Calculate $c_1^{(\ell)}, c_2^{(\ell)}$ (using (6.17)).
 - 6: $\tilde{B}^{(\ell+1)} = \tilde{B}^{(\ell)}, c_1^{(\ell+1)} = c_1^{(\ell)}, c_2^{(\ell+1)} = c_2^{(\ell)}$.
 - 7: **end if**
 - 8: **while** $\max\{\|u^{(\ell)} - u^{(\ell-1)}\|, \|v^{(\ell)} - v^{(\ell-1)}\|\} > \delta$ **do**
 - 9: Set fitting term as $\lambda f(x) = \lambda \left(\left(z - \tilde{B}^{(\ell)} c_1^{(\ell)} \right)^2 - \left(z - \tilde{B}^{(\ell)} c_2^{(\ell)} \right)^2 \right)$.
 - 10: Compute $\rho^{(\ell)}$, (using (3.37)), as discussed in Section 3.3.1 and [18].
 - 11: Compute $u^{(\ell)}$ (using (3.35)) and $v^{(\ell)}$ (using (3.38)).
 - 12: **end while**
 - 13: **end for**
 - 14: $\tilde{B}^* = \tilde{B}^{(\ell)}, c_1^* = c_1^{(\ell)}, c_2^* = c_2^{(\ell)}, \Omega_1^* = u^{(\ell)} > \epsilon, \Gamma^* = \partial\Omega_1^*$.
-

6.3.4 Experimental Results

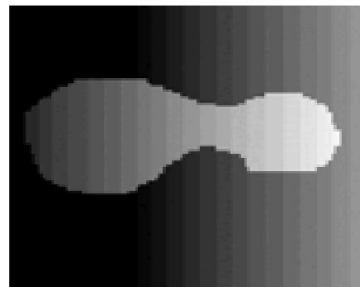
This section is in two parts. First we will test SBF using images from the VMS tests in D. Chen et al. [37], intending to show that the proposed method retains the segmentation quality of VMS, whilst demonstrating the convergence of the intensity constants. Another aspect of the success of SBF is what c_1^* and c_2^* are; we can check whether the computed values are feasible, i.e. $c_1, c_2 \in [0, 1]$, whilst maintaining the quality of the segmentation. Secondly, we investigate other advantages of SBF over VMS. In particular, we look at the segmentation accuracy depending on the fitting parameter λ , and how the piecewise-smooth approximations w^{SBF} and w^{VMS} compare for a model example.

In Test Set 1 we select σ as the value used by D. Chen et al. [37] which is given for the following examples: Image 1 ($\sigma = 3$), Image 2 ($\sigma = 3$), Image 3 ($\sigma = 4$), and Image 4 ($\sigma = 3$). The choice of λ is also related to each example in the same way as other segmentation problems of a similar type [30, 33]. One notes that the selections of λ here do not agree with [37] due to the range of the image intensities, and their size. The values used in the experiments are given as follows: Image 1 ($\lambda = 8$), Image 2 ($\lambda = 40$), Image 3 ($\lambda = 10$), and Image 4 ($\lambda = 5$). In Test Set 2 we vary the fitting parameter $\lambda \in [0, 300]$ and fix $\sigma = 5$, as demonstrated in Figs. 6.5 and 6.6. Further details about parameter selection are discussed in the following section.

Test Set 1: Convergence Behaviour



i) Image 1



ii) Image 2



iii) Image 3



iv) Image 4

Figure 6.2: Images tested with SBF and compared to results of D. Chen et al. [37].

We test four examples (Images 1-4) in Fig. 6.2, all used in D. Chen et al. [37]. In Fig. 6.3, we present the results for each case. We set $\mathbf{s} = 1$, and vary the constraint parameter for each case. Below we give the $\|\tilde{B}\|_F$ value computed for each example tested, in order to ascertain to what extent the bias field is stabilised. Given this value for \mathbf{s} we expect $\|\tilde{B}\|_F \approx 128$ (for a 128×128 image) with the addition of the constraint. We also provide the computed intensity constants, c_1^* and c_2^* , which should be in $[0, 1]$ in order to be consistent with the image. By providing these values, we hope to contextualise the results in relation to the image model and the proposed formulation. It is worth noting that in some sense the value of \mathbf{s} is arbitrary as the constraint is also dependent on γ . Experimentally, similar results as presented below can be attained for different values of \mathbf{s} , if γ is adjusted accordingly. However, setting $\mathbf{s} = 1$ and varying γ is the most intuitive approach to take.

For Image 1 $\gamma = 0.1$, and the intensity constants converge to $c_1^* = 0.352, c_2^* = 0.240$. The bias field, \tilde{B} , also converges and we compute $\|\tilde{B}\|_F = 129.8$. This result is of a similar quality to VMS, shown in Fig. 1 of [37]. For Image 2 $\gamma = 0.2$, and the intensity constants converge to $c_1^* = 0.807, c_2^* = 0.660$. The bias field, \tilde{B} , also converges and we compute $\|\tilde{B}\|_F = 67.0$. However, row 2 of Fig. 6.3 demonstrates that the convergence of c_1 and c_2 is quite slow, taking over 500 iterations which is much more than for the convergence of w_{VMS} in VMS. Image 3 is an ultrasound image, containing intensity variation in the background. For this example $\gamma = 0.1$, and the intensity constants converge to $c_1^* = 0.446, c_2^* = 0.383$. The bias field, \tilde{B} , also converges and we compute $\|\tilde{B}\|_F = 120.9$. Image 4 is another example of vessel segmentation, where intensity varies smoothly throughout the vessel. For this example $\gamma = 0.1$, and the intensity constants converge to $c_1^* = 0.352, c_2^* = 0.239$. The bias field, B , also converges and we compute $\|\tilde{B}\|_F = 122.2$. We see results of comparable quality to rows 3 and 4 in Fig. 6.3, in Figs. 9 and 4 of [37] respectively, except that there is no convergence for c_1 and c_2 . This demonstrates a clear improvement for these examples. Despite the slow convergence of SBF in the case of Image 2, we have fast convergence in the other examples, meaning the additional constraint generally doesn't slow down the computation of a solution. Also, the results are not sensitive to the constraint parameter, γ . For Image 2, it was adjusted to 0.2, but for all other cases it was set at 0.1, and for all examples, $c_1^*, c_2^* \in [0, 1]$, showing that the method produces results consistent with the image.

In Fig. 6.4 we see the piecewise-smooth Mumford-Shah approximations of Images 1-4, using the SBF formulation. Visually these are similar to the corresponding VMS approximations (not shown here). This is to be expected as the advantages of SBF are primarily expected to be in the convergence of the intensity constants c_1 and c_2 . The images tested (particularly 1,2, and 4) have considerable amounts of intensity inhomogeneity, such that they are clearly beyond the piecewise-constant framework of Chan-Vese [33]. The converged intensity constants are very close to one another, such that without a bias field function (or alternative consideration [82, 34, 100]), an accurate segmentation would not be possible. In the next set of results we compare

SBF and VMS explicitly to consider additional advantages of the proposed formulation.

Test Set 2: Comparison to VMS

With Test Set 1, we have successfully demonstrated that SBF achieves the intended goal: the convergence of the intensity constants within a feasible range, and the computation of a stabilised bias field. However, we now intend to examine the success of the proposed method in another way: how does the method affect the accuracy of the final segmentation. With this in mind, we can quantifiably measure the solution of each model (VMS and SBF) against this using the Tanimoto Coefficient [48] as in Chapter 4:

$$TC = \frac{N(GT \cap \Omega_1^*)}{N(GT \cup \Omega_1^*)}, \quad (6.18)$$

where $N(\cdot)$ is the number of pixels in the enclosed region, GT is the ground truth, and Ω_1^* is the result computed with VMS or SBF. However, without the ground truth data for Images 1-4 (Fig. 6.2) we cannot measure this for the examples used by D. Chen et al. [37]. Instead, we test one model example shown in Fig. 6.5 where the ground truth is known precisely. We observe two things with this example. Firstly, how w^{VMS} and w^{SBF} compare visually with each other. In Fig. 6.5 we can see that around Γ^* there are significant differences between the two approximations. SBF appears to produce a sharper approximation of the image, dealing with the discontinuity in the intensity more effectively. Secondly, we have tested each model with a large range of the fitting parameter λ . Our results are promising in the sense that an optimal result can be computed for a wider range of λ with SBF over VMS, as highlighted in Fig. 6.6. It is worth noting that we have not observed such pronounced results with Images 1-4, although an advantage is still present.

6.4 Selective Segmentation with SBF

As discussed in Chapter 5 and the references therein, selective segmentation is the task of extracting one particular object of interest, from a foreground with similar characteristics. We now consider the problem of selecting objects in images that contain significant intensity inhomogeneity, which is beyond recent work on selective segmentation [105, 140]. By incorporating the proposed SBF idea into a current selective segmentation model we aim to demonstrate the flexibility of SBF as a fitting term. We now reintroduce the selective term from the previous chapter and apply it to SBF. The formulation is given as follows:

$$E^{CDSS}(\Gamma, c_1, c_2) = |\Gamma| + \lambda \int_{\Omega} ((z - c_1)^2 \chi_1 + (z - c_2)^2 \chi_2) dx + \theta \int_{\Omega} P_d(x) \chi_1 dx, \quad (6.19)$$

where $P_d(x)$ is the normalised Euclidean distance of each point $x \in \Omega$ from its nearest point in the user-defined polygon. Further details are also given in [112]. Whilst results demonstrate this approach is robust, even in quite difficult cases, it is limited by the

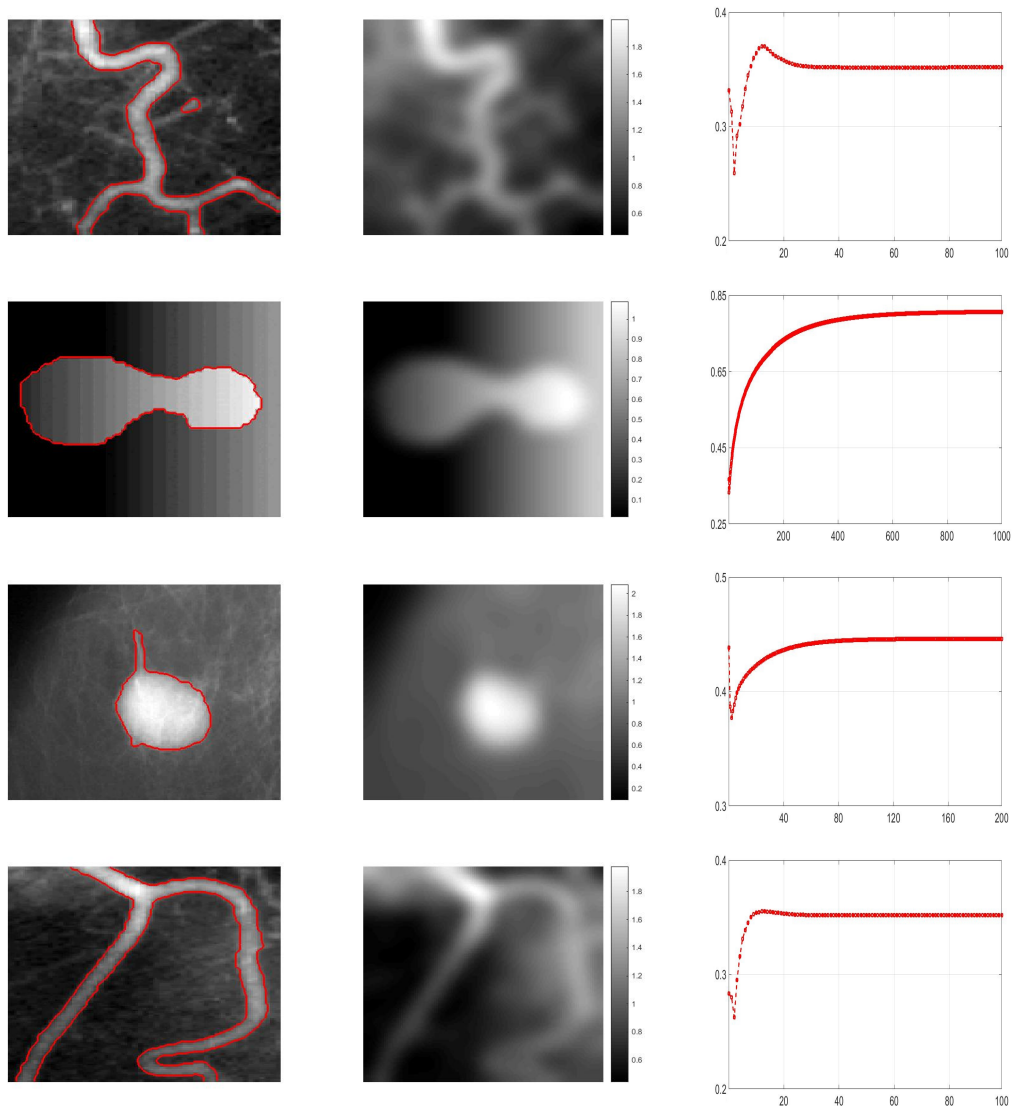
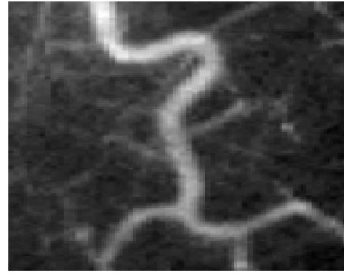
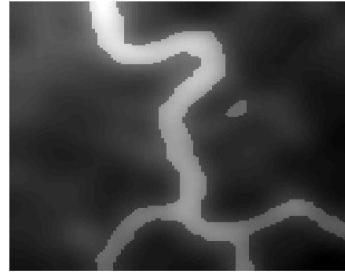


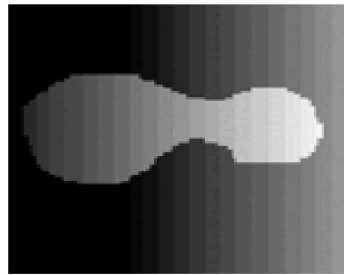
Figure 6.3: SBF Set 1 Results. SBF convergence behaviour (see Section 6.3.4.) Rows 1-4 are for Images 1-4 respectively. From left to right: $z(x)$ and Γ^* computed with SBF, bias field $\tilde{B}(x)$, and the progression of c_1 values (vertical axis) against iterations (horizontal axis.) Similar behaviour for c_2 values is also observed.



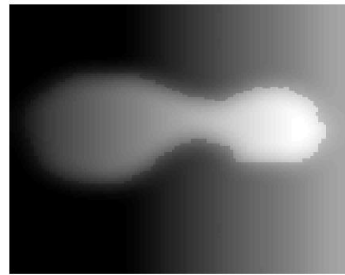
i) Image 1, $z(x)$



ii) Image 1, $w^{SBF}(x)$



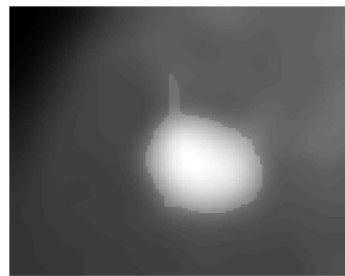
ii) Image 2, $z(x)$



ii) Image 2, $w^{SBF}(x)$



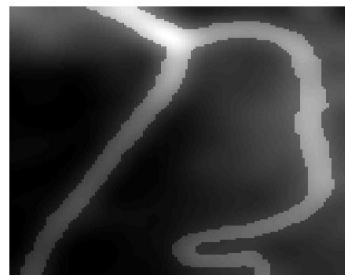
iii) Image 3, $z(x)$



ii) Image 3, $w^{SBF}(x)$



iv) Image 4, $z(x)$



ii) Image 4, $w^{SBF}(x)$

Figure 6.4: SBF Set 1 Results. Piecewise-smooth Mumford-Shah approximation of Images 1-4, given from top to bottom. On the left is z , and on the right is the SBF piecewise-smooth Mumford-Shah approximation, given by w^{SBF} (6.13).

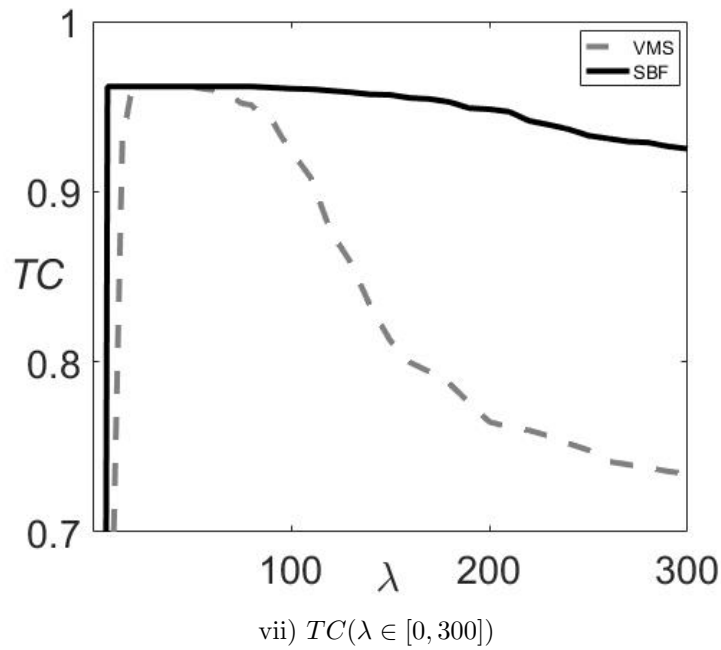
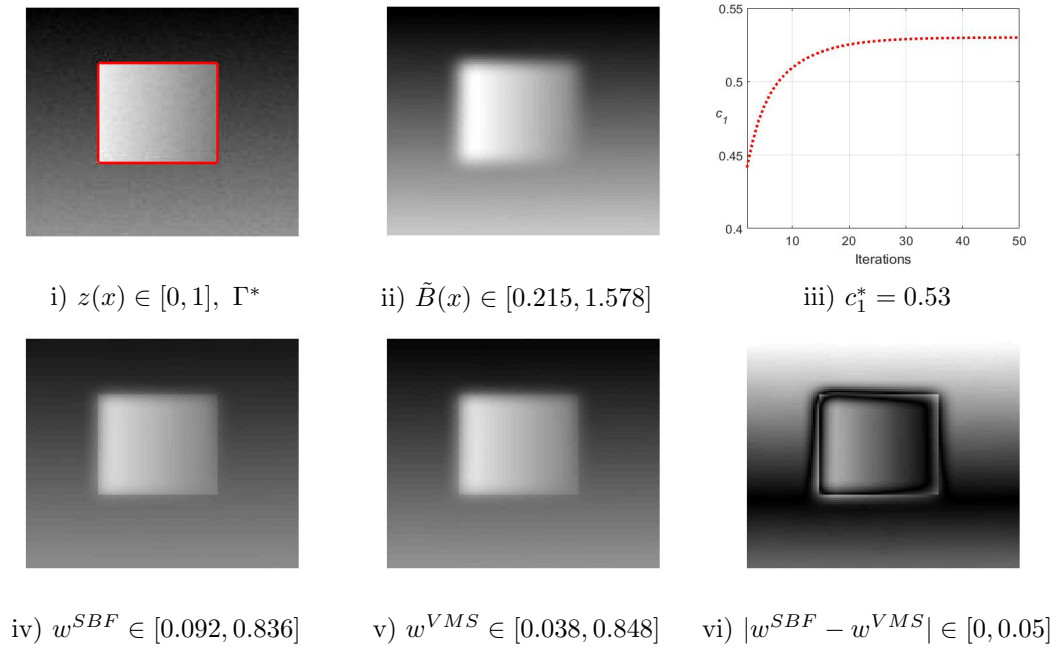


Figure 6.5: SBF Set 2 Results. SBF compared to VMS (see Section 6.3.4.) i) Successful segmentation of the image, $z(x)$, given by contour Γ^* . ii) Computed stabilised bias field, $\tilde{B}(x)$. iii) Convergence of c_1 values (50 iterations.) Similar behaviour for c_2 values is also observed. iv) Piecewise-smooth approximation of $z(x)$ with SBF. v) Piecewise-smooth approximation of $z(x)$ with VMS. vi) Difference between SBF and VMS approximations, demonstrating significant differences around Γ^* . vii) The TC measure for VMS (dotted red) and SBF (blue), demonstrating the segmentation quality falls away for VMS with large values of λ .

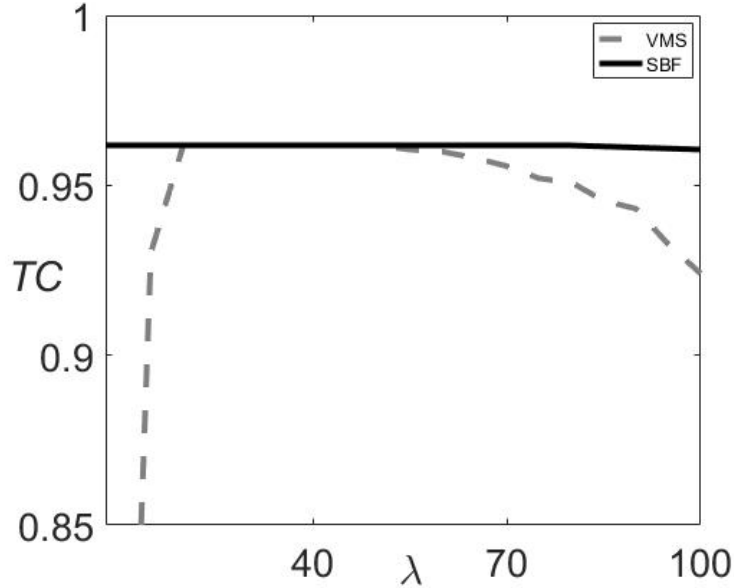


Figure 6.6: SBF Set 2 Results. SBF compared to VMS (see Section 6.3.4.) The TC measure for $\lambda \in [5, 100]$ shows that an optimal Γ^* can be computed for a larger range of λ with SBF than VMS.

piecewise-constant assumption it relies on. We therefore extend this idea to incorporate bias field estimation, which we call Selective SBF:

$$\begin{aligned}
 E^{SSBF}(\Gamma, c_1, c_2, \tilde{B}) = & |\Gamma| + \lambda \int_{\Omega} \left((z - \tilde{B}c_1)^2 \chi_1 + (z - \tilde{B}c_2)^2 \chi_2 \right) dx \\
 & + \mu \int_{\Omega} |\nabla \tilde{B}|^2 dx + \gamma \int_{\Omega} (\tilde{B} - \mathbf{s})^2 dx + \theta \int_{\Omega} P_d(x) \chi_1 dx. \quad (6.20)
 \end{aligned}$$

We minimise this functional (6.20) as outlined in Sections 6.3.2 and 6.3.3 above, except that for Step (3) we use the improved additive operator splitting method from Chapter 4, as we are no longer comparing our results to VMS.

6.4.1 Experimental Results

For Selective SBF we test one image that involves significant intensity inhomogeneity in the foreground and background, shown in Fig. 6.7. The foreground consists of a series of distinct objects that could conceivably be of interest, and was chosen as it is clearly beyond the scope of the piecewise-constant framework of CDSS used in Chapter 5 and [112]. By using just four markers to loosely define the shape of the target object, as well as its location and size, we define a distance selection term P_d that is capable of excluding unwanted artefacts. We demonstrate that we get a successful result for this example, both in terms of the computed contour Γ^* , and the convergence of the intensity constants c_1 and c_2 . A particularly challenging aspect of this type of problem can be highlighted by noting that the intensity constants computed are very close: $c_1^* = 0.35$, and $c_2^* = 0.33$. The role of the stabilised bias field, \tilde{B} , is particularly important here.

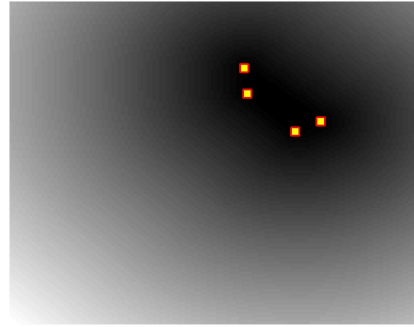
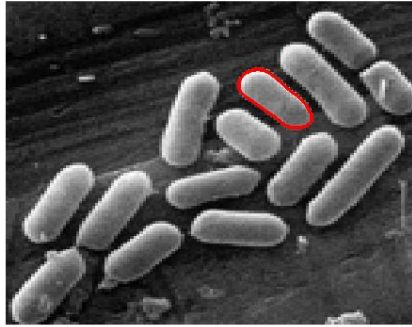
It is worth considering two alternatives for this example, that we now briefly discuss.

Firstly, what would the performance of CDSS be like for the image in Fig. 6.7? For brevity, we do not include those results here. As might be expected for a model that relies on a piecewise-constant framework, the results for this image are inadequate as the segmentation favours exterior artefacts to the target object that are of a similar intensity value. Secondly, what does SBF contribute here, i.e. what would Selective VMS ($\gamma = 0$ in (6.20)) results be like? Again, we do not include results here, but Selective VMS is capable of achieving a successful segmentation, although as expected c_1 and c_2 do not converge. However, we observe a similar effect as observed in Results Set 2 given by Fig. 6.5. That is, with all other parameters fixed and varying the selection parameter θ , there is a successful result for a wider range of values. We do not know the ground truth for this case, which makes quantifying differences between methods difficult, but we aim to further investigate this phenomenon with different examples.

6.5 Remarks

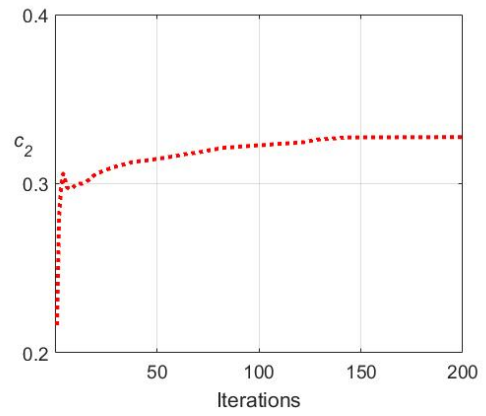
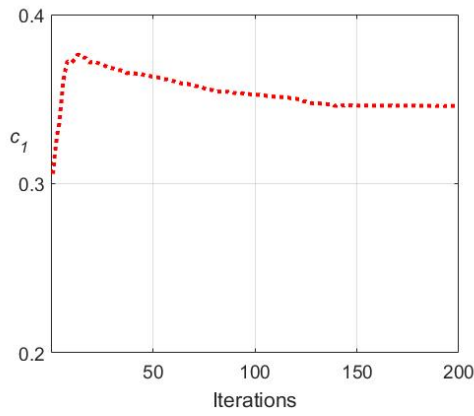
We have proposed the introduction of a constraint to the Variant Mumford-Shah Model [37], although it applies to any model using bias field correction in this way. It is a framework that provides a link between the Mumford-Shah functional [89] and the piecewise-constant functional of Chan and Vese [33], as discussed in Section 6.3.1. This constraint does not affect the computation time as we have shown how the exact minimiser can be well approximated in a similar way to D. Chen et al. [37]. It is an improvement over current methods in the sense that the intensity constants reliably converge and are feasible in relation to the image. This allows for a meaningful representation of the data by the definition of the image model (6.3). We also observe possible advantages with this framework in terms of the quality of the piecewise-smooth approximation of the image, and a model less reliant on the fitting parameter. We have successfully extended the proposed method to a selective segmentation model, incorporating the distance selection term from the previous chapter, to allow for selection in the presence of intensity inhomogeneity, and have again observed an improvement in terms of parameter dependence. This is a potentially important finding, as this 'stabilisation' of the bias field appears to allow for more parameter variation thus improving the reliability of the models. We will investigate this idea further in the future, and attempt to accurately quantify an improvement.

It is important to note here that in the numerical implementation of SBF, detailed in Section 6.3.3, we were consistent with the method of D. Chen et al. [37]. In particular, we used the dual formulation of Chambolle [23, 18] rather than the AOS methods proposed in Chapter 4. Whilst we observed improvements with AOS for GCS over the dual formulation, we wanted a fair comparison between SBF and VMS. This involved obtaining a solution for the VMS case in a way faithful to the approach in [37]. Additionally, it seemed appropriate to repeat this when obtaining a solution in the SBF case. In this way the conclusions drawn from the proposed method are well founded.



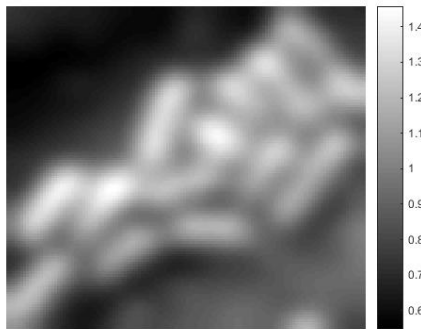
i) $z(x) \in [0, 1], \Gamma^*$

ii) $P_d(x) \in [0, 1]$



iii) $c_1^* = 0.35$

iv) $c_2^* = 0.33$



v) $\tilde{B}(x) \in [0.546, 1.458]$

vi) $u(x) \in [0, 1]$

Figure 6.7: Selective SBF Results (see Section 6.4.1.) i) Successful selective segmentation of the image, $z(x)$, given by contour Γ^* . ii) Distance selection function, $P_d(x)$, with user markers. iii) Convergence of c_1 values (200 iterations.) iv) Convergence of c_2 values (200 iterations.) v) Computed stabilised bias field, $\tilde{B}(x)$. vi) Computed indicator function, $u(x)$.

Chapter 7

Simultaneous Reconstruction and Segmentation

7.1 Introduction

In this chapter we consider the situation when forming a fitting function based on the observed image is not possible due to insufficient quality of the data. Specifically, we aim to segment images that contain blur. In this case the observed image must be reconstructed in order to segment it accurately. Several recent methods to combine image segmentation and deconvolution in the case where the blur function k is known or of known type have been proposed, but in the case where the blur function is unknown there have not been many significant advances. Here, we propose two variational models for the simultaneous reconstruction and segmentation of blurred images with spatially invariant blur in the blind case (i.e. the blur type is unknown).

Variational segmentation models based on intensity similarity often employ edge detection techniques to aid the segmentation and some can handle fuzzy boundaries [13, 106]. Generally these approaches can deal with the presence of noise, but blur is more problematic and most variational models are incapable of obtaining accurate results, particularly in cases where there is a reliance on the edge detector. Work in the segmentation of blurred images is at an early stage but there exist methods, such as those presented in [11, 28, 107, 69], which use the framework of Mumford-Shah [33, 5, 74] or Chan-Vese [43, 42] and TV image restoration [109, 124].

The main contribution of this chapter is the proposal of two models that incorporate blind deconvolution (with implicitly constrained image reconstruction) and globally convex segmentation (GCS). The former offers advantages over hard constraints such as scaling or truncation, whilst the latter enables us to compute global minimisers for two-phase segmentation models with a fixed fitting function. In a similar way to Bar et al. [11], we form a joint functional and minimise it alternately in order to simultaneously reconstruct and segment the image. We also present a relaxed method for improved accelerated convergence. Our tests demonstrate that related models that do not impose constraints for the restoration (and the restored blur kernel in the blind case) do not perform as well as our proposed method, especially for heavy blur.

This chapter is organised as follows. In Section 7.2, we review existing approaches

to segmenting blurred images and introduce relevant image reconstruction methods. In Section 7.3.1, we introduce two new two-stage models for the cases of images in the presence of Gaussian noise and Poisson noise incorporating implicitly constrained deblurring and GCS. The primary purpose of these models is to examine the advantages of reconstructing and segmenting simultaneously and in Section 7.3 we introduce our main contribution; a joint model for the segmentation of blurred images. In Section 7.4, we introduce our relaxed model using alternate direction methods for accelerated convergence. In Section 7.5, we present experimental results. Finally, in Section 7.6 we offer some concluding remarks on this work. We note here that we make an adjustment to the conventional notation. In the following we will refer to the image function as $v(x)$, as opposed to $u(x)$ (as it is most commonly in the literature). This is in order to avoid confusion between the reconstructed image function, and the segmentation function as defined in previous chapters.

7.2 Existing Methods

Approaches to the problem of segmenting blurred images can be split into two categories: two-stage and joint. In the first the aim is to reconstruct the true image, followed by a segmentation step [28, 107]. For joint approaches, reconstruction and segmentation is carried out simultaneously by minimising a joint functional [11, 69]. In this section, we review some relevant examples of each approach.

Throughout this work we assume that the blur is spatially invariant, such that the observed image, z , may be modeled as the convolution of the true image v with a point spread function k , with the possibility of some additive noise:

$$z(x) = (k * v)(x) + \eta(x), \quad (7.1)$$

where the operation of convolution is denoted by $*$ and is defined in Chapter 3.

7.2.1 Segmentation of Blurred Images

In 2004, Bar et al. [11] coupled the segmentation and reconstruction processes into a joint functional, considering both non-blind and semi-blind deconvolution (where σ is to be found, for Gaussian blur). They minimised the joint functional

$$f_{BSK}(v, k_\sigma, g) = \frac{1}{2} \int_{\Omega} (k_\sigma * v - z)^2 dx + \beta \int_{\Omega} g^2 |\nabla v|^2 dx + \alpha \int_{\Omega} \varepsilon |\nabla g|^2 + \frac{(g-1)^2}{4\varepsilon} dx + \gamma \int_{\Omega} |\nabla k_\sigma|^2 dx, \quad (7.2)$$

which is dependent on the image v , the edge integration map g and the kernel function parameter σ . A special case exists in the case of known blur where minimisation with respect to the kernel width is not necessary, and neither is the final term of the functional. Minimising with respect to the arguments simultaneously reconstructs and segments the observed image in either the non-blind or semi-blind case.

In 2009 Jung et al. [69] introduced a model for two-phase segmentation of blurred images by incorporating the Chan-Vese [33] formulation into a deblurring term as follows:

$$f_{JCSV}(c_1, c_2, \phi) = \int_{\Omega} |z - k * (c_1 H(\phi) + c_2 (1 - H(\phi)))|^2 dx + \mu \int_{\Omega} |\nabla H(\phi)| dx$$

They also extended this idea to multi-phase segmentation and denoising, but this form is of particular interest to our work. In [97], the authors address formulating this model in a convex relaxation framework. The authors minimise the functional above alternately with H^1 gradient descent.

Another approach, this time in the blind case, is from Reddy et al. [107]. They incorporate the unknown restored image in the Chan-Vese formulation as follows:

$$f_{RCR} = \mu |\Gamma| + \nu \text{Area}(\text{in}(\Gamma)) \\ + \lambda_1 \int_{\text{in}(\Gamma)} |v(x) - c_1|^2 dx + \lambda_2 \int_{\text{out}(\Gamma)} |v(x) - c_2|^2 dx,$$

where $v(x) = |k(x) * z(x)|^2$ is the square of the convolution of the observed data z and a Gaussian kernel. The c_1, c_2 and Γ terms and the parameters are consistent with the original Chan-Vese functional [33], as detailed in Chapter 3. The authors employ a two-stage approach, that amounts to segmenting the reconstructed image.

In 2014 Chan et al. [28] proposed a two-stage convex method for the segmentation of blurred images, corrupted by either Poisson or multiplicative Gaussian noise. Their approach is to extract a smooth image v from the observed image z and threshold v in such a way that the segmentation regions are defined. Given the blurring operator \mathcal{A} , the functional is given as

$$f_{CZY}(v) = \int_{\Omega} |\nabla v| dx + \frac{\mu}{2} \int_{\Omega} |\nabla v|^2 dx + \lambda \int_{\Omega} \mathcal{A}v - f \log \mathcal{A}v dx.$$

Minimising with respect to v has a unique solution and can be solved by Split Bregman [59] or Chambolle-Pock [26].

In the following sections of this chapter we consider both the joint and two-stage approaches using GCS and implicitly constrained deblurring in an attempt to improve the accuracy of the results. First, we review the relevant image reconstruction and segmentation techniques in the non-blind case, i.e. the blur is known.

7.2.2 Two-Stage Approach for Images with Known Blur

We now discuss the details of applying two basic methods to this problem, in a simple two-stage approach. Specifically, we consider segmentation for blurred images in the non-blind case. We first briefly review how to restore an image in this case, and then applying a nonconvex segmentation method to the result. The first stage is to restore the image, employing a ROF-type minimisation problem [109]:

$$\min_{v(x)} \left\{ \int_{\Omega} ([k * v](x) - z(x))^2 dx + \alpha \int_{\Omega} |\nabla v(x)| dx \right\}, \quad (7.3)$$

where $k(x)$ is the known point spread function describing the blur degradation of the true image, $z(x)$ is the observed blurred and noisy image, $v(x)$ is the unknown image which is to be restored, and α is a parameter which balances between data fitting and regularisation. The function $v(x)$ which minimises the functional of (7.3) is the restored image, in which a segmentation step such as Chan-Vese [33] is then applied. Further details of the Chan-Vese step can be found in Chapter 3. This technique may achieve a good result, however, the blur function must be already known and the segmentation result of the nonconvex Chan-Vese problem is heavily dependent on the restored image, v .

In the next section we extend this approach to the blind case, where the blur function is unknown. We incorporate transformations which allow for the image intensities to be constrained implicitly, which will be discussed in the next section, and formulate it in a GCS framework introduced in previous chapters.

7.3 Segmentation of Images Corrupted By Unknown Blur

This section concerns segmenting images with unknown blur, which we call blind image segmentation. First, we discuss a two-stage approach that aims to improve on the method described in the previous section and the existing methods in the literature [11, 69]. We then introduce a model which simultaneously restores and segments the image, by proposing a joint functional.

7.3.1 Two-Stage Approach

Based on the existing methods reviewed in the previous section, we propose a two-stage model for segmentation with blind deconvolution. That is, we first restore the sharp image from the corrupted observed data without knowledge of the blur function and then segment the result. This is a conventional approach, with examples including work by Reddy et al. [107] and Chan et al. [28].

Beginning with blind deconvolution, we attempt to restore the image and blur function simultaneously. Following similar approaches [135, 35] we deblur the observed image by solving the regularised joint minimisation problem

$$\arg \min \left\{ f_{CW}(v, k) = \int_{\Omega} ([k * v](x) - z(x))^2 dx + R_1(v(x)) + R_2(k(x)) \right\}, \quad (7.4)$$

where R_1 and R_2 are regularisation terms that enforce the smoothness constraints on the image and blur function respectively. The minimisation of (7.4) is subject to the constraints

$$k(x) \geq 0, \quad v(x) \geq 0, \quad \int_{\Omega} k(x) dx = 1, \quad k(x) = k(-x), \quad (7.5)$$

and are imposed explicitly at each outer iteration of an alternate minimisation scheme. These are imposed in order to find a unique solution of this jointly nonconvex problem. Chen et al. [38] proposed an improved method, enforcing the non-negative constraints implicitly. Alternative approaches, where the intensity values are projected back onto the correct range, can lead to a significant reduction in the quality of the recovered image [27]. We now briefly discuss a transform $\tau_{\mathbf{a}}(\psi)$ where $\tau_{\mathbf{a}} : \mathbb{R}^{m \times n} \rightarrow \mathcal{C}$, $\mathcal{C} \subseteq \mathbb{R}^{m \times n}$ is a function with a range constrained in \mathcal{C} , and $\psi : \Omega \rightarrow \mathbb{R}^{m \times n}$ is a function such that $\tau_{\mathbf{a}}(\psi)$ is equal to the image v . The proposed bounded transform is given in [27] by

$$\tau_{\mathbf{a}}(\psi) = \frac{a_1 + 2a_4}{1 + a_2 e^{-\frac{2\psi}{a_3}}} - a_4, \quad (7.6)$$

where the parameters a_1, a_2, a_3, a_4 are problem dependent. Further details about this method can be found in [38] and related literature. The inverse transform $\psi : \mathcal{C} \rightarrow \mathbb{R}^{m \times n}$ is given such that $\psi = \xi_{\mathbf{a}}(v)$. To avoid introducing non-linearity in the fitting term, the augmented Lagrangian method [50, 132] is employed through the term $\mathcal{A}_{\mathbf{a}}$. The point spread function is treated in a similar way with a transform $\tau_{\mathbf{a}}(k)$, for parameters b_1, b_2, b_3, b_4 , and a Lagrangian term $\mathcal{A}_{\mathbf{b}}$.

With these considerations in mind, the authors [38] introduced the following non-negative implicitly constrained functional:

$$\begin{aligned} f_{CHWZ}(v, k, \psi, \omega) = & \int_{\Omega} (k * v - z)^2 dx + R_1(\tau_{\mathbf{a}}(\psi)) + R_2(\tau_{\mathbf{b}}(\omega)) \\ & + \mathcal{A}_{\mathbf{a}}(v, \psi; \varphi_1, \gamma_1) + \mathcal{A}_{\mathbf{b}}(k, \omega; \varphi_2, \gamma_2). \end{aligned} \quad (7.7)$$

Here R_1 and R_2 denote the TV regularisation for the image and blur function respectively:

$$R_1(\tau_{\mathbf{a}}(\psi)) = \int_{\Omega} |\nabla \tau_{\mathbf{a}}(\psi)| dx, \quad R_2(\tau_{\mathbf{b}}(\omega)) = \int_{\Omega} |\nabla \tau_{\mathbf{b}}(\omega)| dx,$$

and \mathcal{A} penalises the distance between the image (blur) function and transformed ψ (ω) functions respectively, which are given by

$$\mathcal{A}_{\mathbf{a}}(v, \psi; \varphi_1, \gamma_1) = \gamma_1 \int_{\Omega} (v - \tau_{\mathbf{a}}(\psi))^2 dx + \langle \varphi_1, v - \tau_{\mathbf{a}}(\psi) \rangle, \quad (7.8)$$

$$\mathcal{A}_{\mathbf{b}}(k, \omega; \varphi_2, \gamma_2) = \gamma_2 \int_{\Omega} (k - \tau_{\mathbf{b}}(\omega))^2 dx + \langle \varphi_2, k - \tau_{\mathbf{b}}(\omega) \rangle. \quad (7.9)$$

Alternate minimisation of the functional (7.7) is achieved by solving the resulting Euler-Lagrange equations

$$\begin{aligned} \mathcal{E}_{CHWZ}^1(v, k, \psi, \varphi_1) &= 0, & \mathcal{E}_{CHWZ}^2(v, \psi) &= 0, \\ \mathcal{E}_{CHWZ}^1(k, v, \omega, \varphi_2) &= 0, & \mathcal{E}_{CHWZ}^2(k, \omega) &= 0, \end{aligned}$$

where the functions \mathcal{E}_{CHWZ}^1 and \mathcal{E}_{CHWZ}^2 are given by

$$\mathcal{E}_{CHWZ}^1(v, k, \psi, \varphi_1) = k^\dagger * (k * v - z) + \gamma_1 (v - \tau_{\mathbf{a}}(\psi)) + \varphi_1, \quad (7.10)$$

$$\begin{aligned} \mathcal{E}_{CHWZ}^2(v, \psi) &= \alpha_1 \frac{\partial \tau_{\mathbf{a}}}{\partial \psi} \nabla \cdot \left(\frac{\nabla \tau_{\mathbf{a}}(\psi)}{|\nabla \tau_{\mathbf{a}}(\psi)|} \right) + \gamma_1 (\tau_{\mathbf{a}}(\psi) - v) \frac{\partial \tau_{\mathbf{a}}}{\partial \psi} \\ &+ \varphi_1 (\tau_{\mathbf{a}}(\psi) - v) \frac{\partial \tau_{\mathbf{a}}}{\partial \psi} - \varphi_1 \frac{\partial \tau_{\mathbf{a}}}{\partial \psi}, \end{aligned} \quad (7.11)$$

where $k^\dagger(x) = k(-x)$. We can solve (7.10) efficiently with Fourier transforms and (7.11) using gradient descent methods. Solutions for k and ω can be found in a similar way. We restore the image by alternately minimising (7.7) until an acceptable tolerance is reached, solving (7.10) and (7.11) to restore the image (v and ψ), followed by updating the dual function φ_1 . A similar procedure is then followed for the blur function (k and ω) and the dual function φ_2 .

Once a solution for v has been computed, we then segment the restored image as opposed to the observed image. We use the convex relaxation framework, discussed in Chapter 4, where a segmentation function $u \in [0, 1]$ is thresholded to produce the final contour. As in Chapter 5 we assume that the true intensity constants are unknown such that the joint segmentation problem is as follows:

$$\min_u \left\{ \mu \int_{\Omega} |\nabla u| dx + \lambda \int_{\Omega} [|v - c_1|^2 u + |v - c_2|^2 (1 - u)] dx + \alpha \int_{\Omega} \nu_{\epsilon}(u) dx \right\}. \quad (7.12)$$

It is possible to adjust the approach slightly to be more consistent with Chapter 4 if reasonable approximations of the intensity constants are available, by fixing c_1 or c_2 . We can derive the Euler-Lagrange equation with respect to u in a similar way as to presented in Chapter 2, giving

$$0 = \mu \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \lambda ((v - c_1)^2 - (v - c_2)^2) - \alpha \nu'_{\epsilon}(u). \quad (7.13)$$

As before this equation can be solved using AOS [85, 129] with the following update, after discretisation and rewriting in the matrix-vector form:

$$u^{n+1} = \frac{1}{2} \sum_{\ell=1}^2 (I - 2\tau \mu A_{\ell}(u^n))^{-1} (u^n + \tau f^n), \quad (7.14)$$

where $f = -\lambda ((v - c_1)^2 - (v - c_2)^2) - \alpha \nu'_{\epsilon}(u)$. One notes that we are using AOS0 from Chapter 4, and improved results could be obtained by considering AOS1. Here, A_{ℓ} is the diffusion quantity in the ℓ direction ($\ell = 1, 2$ for first and second dimensions respectively). Minimising with respect to c_1 and c_2 (keeping other arguments fixed), we get the following equations:

$$c_1(v, u) = \frac{\int_{\Omega} v u dx}{\int_{\Omega} u dx}, \quad c_2(v, u) = \frac{\int_{\Omega} v(1 - u) dx}{\int_{\Omega} (1 - u) dx}. \quad (7.15)$$

We summarise these details in Algorithm 3. An initial estimate of the image is given by the observed image, from which the restored image, v , is found. Then, we segment

this new image using the methods discussed in earlier chapters. This method is in two distinct stages, where the blur is assumed to be Gaussian. We refer to this model as Two-Stage Gaussian (or 2SG), from here.

Algorithm 3 Segmentation of blurred images: $u^{(\ell)} \leftarrow 2SG(u^{(0)}, z, maxit)$

```

1:  $v^{(0)} \leftarrow z, \psi^{(0)} \leftarrow \xi_{\mathbf{a}}(v^{(0)})$ 
2:  $\varphi_1^{(0)} \leftarrow \mathbf{1}$ 
3: for  $\ell \leftarrow 1 : maxit$  do
4:   Update  $v^{(\ell)}$  by solving (7.10)
5:   Update  $\psi^{(\ell)}$  by solving (7.11)
6:   Update  $\varphi_1^{(\ell)} \leftarrow \varphi_1^{(\ell-1)} + \gamma_1 (u^{(\ell)} - \tau_{\mathbf{a}}(\psi^{(\ell)}))$ 
7:   Update  $k^{(\ell)}$ 
8:   Update  $\omega^{(\ell)}$ 
9:   Update  $\varphi_2^{(\ell)} \leftarrow \varphi_2^{(\ell-1)} + \gamma_2 (k^{(\ell)} - \tau_{\mathbf{b}}(\omega^{(\ell)}))$ 
10: end for
11:  $v \leftarrow \tau_{\mathbf{a}}(\psi^\ell)$ 
12: for  $\ell \leftarrow 1 : maxit$  do
13:   Calculate  $c_1^{(\ell)} \leftarrow c_1(v, u^{(\ell-1)})$ ,  $c_2^{(\ell)} \leftarrow c_2(v, u^{(\ell-1)})$  using (7.15)
14:   Update  $u^{(\ell)}$  by solving  $\mathcal{E}_{cs}(u^{(\ell-1)}) = 0$  using (7.13)
15: end for

```

We now consider the case of Poisson noise being present in the image, and make an adjustment to our two-stage algorithm. We thus attempt to restore the true image from the corrupted image by solving the Robust Richardson Lucy problem, employing the function $\Phi(s) = 2\sqrt{s + \beta}$ [98]. Solving this problem, we obtain an approximation of the true image. In this two-stage setting, once we have obtained the approximation of the image, we proceed with the segmentation as described in the Gaussian case above. This is outlined in Algorithm 4 below. It can be noted that while this restoration method provides a restriction on the lower bound of the restored image v , it does not provide an upper limit. We may obtain this by a projection \mathbb{P} of the restored data onto the ideal range at each iteration. We refer to this model as Two-Stage Poisson (or 2SP), from here.

Algorithm 4 Segmentation of blurred images: $u^{(\ell)} \leftarrow 2SP(u^{(0)}, z, maxit)$

```

1:  $v^{(0)} \leftarrow z$ 
2: for  $\ell \leftarrow 1 : maxit$  do
3:   Update the image  $v^{(\ell)}$ 
4:   Update the blur function  $k^{(\ell)}$ 
5:   Update transformed blur function  $\omega^{(\ell)}$ 
6:   Update  $\varphi_2^{(\ell)} \leftarrow \varphi_2^{(\ell-1)} + \gamma_2 (k^{(\ell)} - \tau_{\mathbf{b}}(\omega^{(\ell)}))$ 
7: end for
8:  $v \leftarrow v^\ell$ 
9: for  $\ell \leftarrow 1 : maxit$  do
10:   Calculate  $c_1^{(\ell)} \leftarrow c_1(v, u^{(\ell-1)})$ ,  $c_2^{(\ell)} \leftarrow c_2(v, u^{(\ell-1)})$  using (7.15)
11:   Update  $u^{(\ell)}$  by solving  $\mathcal{E}_{cs}(u^{(\ell-1)}) = 0$  using (7.13)
12: end for

```

We will demonstrate in Test Set 1 in Section 7.5 that deblurring considerations are important for obtaining an accurate segmentation of a blurred image. We also demonstrate that the advanced techniques described here offer improvements compared to similar techniques.

7.3.2 A Joint Model for Blind Image Segmentation

We now consider the simultaneous restoration and segmentation of a blurred image, by constructing a joint variational model. Broadly speaking, there are two approaches for this type of problem in the literature. Firstly, we may replace the image in the deblurring problem with the segmentation function and attempt to restore this while recovering the average intensities. While this may provide good results, [97] has shown that this may not be robust as it is no longer compatible with the convex relaxation framework, and the thresholding procedure cannot obtain an accurate contour reliably. The second approach is to replace the received data term $z(x)$ in the segmentation functional (7.12) by the restored image function $v(x)$ and add the constraint that this function should satisfy the deconvolution minimisation problem (7.4) and associated constraints (7.5). Imposing this constraint by incorporating the terms into the existing functional, we form the new joint minimisation model.

$$\begin{aligned}
f(v, u, c_1, c_2, k) = & \mu \int_{\Omega} |\nabla u| dx + \alpha \int_{\Omega} \nu_{\epsilon}(u) dx \\
& + \lambda \int_{\Omega} [|v - c_1|^2 u + |v - c_2|^2 (1 - u)] dx \\
& + \frac{1}{2} \|k * v - z\|_{L^2(\Omega)}^2 + \alpha_1 \int_{\Omega} |\nabla v| dx + \alpha_2 \int_{\Omega} |\nabla k| dx, \quad (7.16)
\end{aligned}$$

subject to the constraints (7.5). Here, the restored image v provides the intensity and spatial information for the segmentation terms. Based on the approach of implicitly constrained deconvolution discussed in the previous section, we introduce the functions ψ, ω in order to apply constraints on the intensity values of the image and blur functions. We incorporate the transform functions $\tau_{\mathbf{a}}(\psi), \tau_{\mathbf{b}}(\omega)$ into the functional with terms penalising the distance of the image and blur functions from these terms, giving the following minimisation problem:

$$\begin{aligned}
f_{J_1}(u, v, \psi, k, \omega; \varphi_1, \varphi_2) = & \mu \int_{\Omega} |\nabla u| dx + \alpha \int_{\Omega} \nu_{\epsilon}(u) dx + \frac{1}{2} \|k * v - z\|_{L^2(\Omega)}^2 \\
& + \lambda \int_{\Omega} [|\tau_{\mathbf{a}}(\psi) - c_1|^2 u + |\tau_{\mathbf{a}}(\psi) - c_2|^2 (1 - u)] dx \\
& + \alpha_1 \int_{\Omega} |\nabla \tau_{\mathbf{a}}(\psi)| dx + \mathcal{A}_{\mathbf{a}}(v, \psi; \varphi_1, \gamma_1) \\
& + \alpha_2 \int_{\Omega} |\nabla \tau_{\mathbf{b}}(\omega)| dx + \mathcal{A}_{\mathbf{b}}(k, \omega; \varphi_2, \gamma_2),
\end{aligned}$$

subject to unit integral and symmetry constraints on the blur function and where $\mathcal{A}_{\mathbf{a}}$ and $\mathcal{A}_{\mathbf{b}}$ are given by (7.8) and (7.9). Next, we will take each of the arguments in turn

for minimising the joint functional.

Segmentation Step, u, c_1, c_2 :

The model is formulated such that it is consistent with the convex relaxation framework introduced in Chapter 4. As a result, the segmentation step is similar to before using an AOS scheme with

$$f = -\lambda((\tau_a(\psi) - c_1)^2 - (\tau_a(\psi) - c_2)^2) - \alpha\nu'_\epsilon(u).$$

The update for u is then given as:

$$u^{n+1} = \frac{1}{2} \sum_{\ell=1}^2 (I - 2\tau\mu A_\ell(u^n))^{-1}(u^n + \tau f^n). \quad (7.17)$$

The minimisation of the intensity constants is also analogous to previous work:

$$c_1(\psi, u) = \frac{\int_{\Omega} \tau_{\mathbf{a}}(\psi) u dx}{\int_{\Omega} u dx}, \quad c_2(\psi, u) = \frac{\int_{\Omega} \tau_{\mathbf{a}}(\psi)(1-u) dx}{\int_{\Omega} (1-u) dx}. \quad (7.18)$$

Image function v :

Minimising with respect to v gives the equation

$$k^\dagger * (k * v - z) + \gamma_1(v - \tau_{\mathbf{a}}(\psi)) + \varphi_1 = 0, \quad (7.19)$$

which contains the main deconvolution component. This can be rewritten with the left hand side as a convolution of u as

$$[k^\dagger * k + \delta\gamma_1] * v = k^\dagger * z + \gamma_1\tau_{\mathbf{a}}(\psi) - \varphi_1.$$

It is important to note that after the discretisation of this equation, the term $k^\dagger * k$ along with the operation of convolution defines a symmetric positive definite matrix. Put briefly, if \mathbf{k} and \mathbf{v} are the discretised k and v respectively, then we have

$$\bar{\mathbf{k}} \circ \mathbf{k} \circ \mathbf{v} = \mathbf{H}\mathbf{v}$$

for a symmetric positive definite matrix \mathbf{H} where \circ denotes the operation of discrete convolution. We can solve this problem using the preconditioned conjugate gradient method with a preconditioner, following the idea of [125], given by

$$P = (\tilde{k} * \tilde{k} + \chi I)^{\frac{1}{2}} (\chi I + (\gamma_1 + \varphi_1)) (\tilde{k} * \tilde{k} + \chi I)^{\frac{1}{2}}, \quad (7.20)$$

where \tilde{k} is a circulant approximation to k .

Transformed image function ψ :

Minimising with respect to the function ψ , we obtain:

$$\begin{aligned} \mathcal{E}_\psi(\psi) = & 2\lambda_1(\tau_{\mathbf{a}}(\psi) - c_1)\nu\frac{\partial\tau_{\mathbf{a}}(\psi)}{\psi} + 2\lambda_2(\tau_{\mathbf{a}}(\psi) - c_2)(1 - \nu)\frac{\partial\tau_{\mathbf{a}}(\psi)}{\psi} \\ & + \alpha_1\frac{\partial\tau_{\mathbf{a}}(\psi)}{\psi}\nabla\cdot\left(\frac{\nabla\tau_{\mathbf{a}}(\psi)}{|\nabla\tau_{\mathbf{a}}(\psi)|}\right) - \gamma_1(v - \tau_{\mathbf{a}}(\psi))\frac{\partial\tau_{\mathbf{a}}(\psi)}{\psi} - \varphi_1\frac{\partial\tau_{\mathbf{a}}(\psi)}{\partial\psi}. \end{aligned} \quad (7.21)$$

Applying gradient descent, letting $\psi = \psi(x; t)$, we solve the problem

$$\psi_t(x, t) = -\mathcal{E}_\psi(\psi(x, t)) \quad \text{s.t.} \quad \psi_t(x, t)|_{t=0} = \psi^{(0)}(x).$$

Discretising this equation by forward differences in terms of time t and rearranging, we have

$$\psi(x, t + 1) = \psi(x, t) - \tau\mathcal{E}_\psi(\psi(x, t)).$$

Beginning with the initial estimate of ψ at $t = 0$ which is determined by the inverse transform of the received data z in the first instance and the latest approximation in subsequent alternate minimisation iterations, we evolve in time until the stopping criteria is met:

$$\|\psi(x, t + 1) - \psi(x, t)\|_{L^2} = \tau\|\mathcal{E}_\psi(\psi(x, t))\|_{L^2} < \psi_{tol}.$$

Point spread function k :

Minimising with respect to the blur function k , we have the equation for the blur function

$$v^\dagger * (v * k - z) + \gamma_2(k - \tau_{\mathbf{b}}(\omega)) + \varphi_2 = 0, \quad (7.22)$$

which may be solved for k in a similar manner to (7.19)

Transformed point spread function ω :

Finally, minimising with respect to ω , we obtain:

$$\alpha_2\frac{\partial\tau_{\mathbf{b}}(\omega)}{\omega}\nabla\cdot\left(\frac{\nabla\tau_{\mathbf{b}}(\omega)}{|\nabla\tau_{\mathbf{b}}(\omega)|}\right) - \gamma_2(k - \tau_{\mathbf{b}}(\omega))\frac{\partial\tau_{\mathbf{b}}(\omega)}{\partial\omega} - \varphi_2\frac{\partial\tau_{\mathbf{b}}(\omega)}{\partial\omega} = 0, \quad (7.23)$$

which may be solved using a gradient descent scheme in a similar way to solving for ψ above.

Algorithm:

To begin, we use the observed data as the initial estimate for v . We also make an estimate of the point spread function based on visual observation of the received image. Using this information, we obtain the initial estimates of ψ and ω , and compute c_1 and c_2 . We then update the image, v , ψ , the point spread function, k , ω , the segmentation function u , φ_1 and φ_2 , and iterate until we reach an acceptable tolerance. Our algorithm is presented in Algorithm 5. We refer to this model as the Joint Reconstruction and Segmentation (or JRS) model from here.

Algorithm 5 Segmentation of blurred images: $u^{(\ell)} \leftarrow \mathbf{J}(u^{(0)}, k^{(0)}, z, \text{maxit})$

- 1: $v^{(0)} \leftarrow z, \psi^{(0)} \leftarrow \xi_{\mathbf{a}}(v^{(0)}), \omega^{(0)} \leftarrow \xi_{\mathbf{b}}(k^{(0)})$
 - 2: $\varphi_1^{(0)} \leftarrow \mathbf{1}, \varphi_2^{(0)} \leftarrow \mathbf{1}$
 - 3: **for** $\ell \leftarrow 1 : \text{maxit}$ **do**
 - 4: Calculate $c_1^{(\ell)} \leftarrow c_1(\psi^{(\ell-1)}, u^{(\ell-1)}), c_2^{(\ell)} \leftarrow c_2(\psi^{(\ell-1)}, u^{(\ell-1)})$ using (7.18)
 - 5: Update $v^{(\ell)}$ by solving (7.19)
 - 6: Update $\psi^{(\ell)}$ by solving (7.21)
 - 7: Update $k^{(\ell)}$ by solving (7.22)
 - 8: Update $\omega^{(\ell)}$ by solving (7.23)
 - 9: Update $u^{(\ell)}$ by solving (7.17)
 - 10: Update $\varphi_1^{(\ell)} \leftarrow \varphi_1^{(\ell-1)} + \gamma_1(v^{(\ell)} - \tau_{\mathbf{a}}(\psi^{(\ell)}))$
 - 11: Update $\varphi_2^{(\ell)} \leftarrow \varphi_2^{(\ell-1)} + \gamma_2(k^{(\ell)} - \tau_{\mathbf{b}}(\omega^{(\ell)}))$
 - 12: **end for**
-

We demonstrate in Test Set 2 of Section 7.5 that segmenting a blurred image with the JRS model can offer improved results over the corresponding two-stage method. Furthermore, in Test Set 3, we show that our method offers improved results over other comparable methods. In the following section, we consider an alternative joint method which aims to improve the speed of obtaining a solution.

7.4 A Relaxed Model for Blind Image Segmentation

In the joint functional introduced in the previous section, solving for the transformed image function, ψ , is the most time consuming aspect of the minimisation procedure. In this section we consider a way to simplify the equation for ψ by relaxing the functional, aiming to speed up the restored segmentation. To do this we introduce a new variable $\varpi(x)$, which should be equal to the reconstructed image at convergence. In order to do this we include distance measures that drive v and ϖ close together, as well as penalisation parameters to control their influence on the solution. The new problem is given as follows:

$$\begin{aligned}
& \max_{\varphi_1, \varphi_2, \zeta} \min_{u, c_1, c_2, \varpi, v, \psi, k, \omega} \left\{ f_{J_2}(u, c_1, c_2, \varpi, v, \psi, k, \omega; \varphi_1, \varphi_2, \zeta) = \mu \int_{\Omega} |\nabla u| dx \right. \\
& + \lambda \int_{\Omega} [|\varpi - c_1|^2 u + |\varpi - c_2|^2 (1 - u)] dx + \alpha \int_{\Omega} \nu_{\epsilon}(u) dx \\
& + \frac{1}{2} \|k * v - z\|_{L^2(\Omega)}^2 + \frac{v}{2} \|\varpi - \tau_{\mathbf{a}}(\psi)\|_{L^2(\Omega)}^2 + \langle \zeta, \varpi - \tau_{\mathbf{a}}(\psi) \rangle \\
& + \alpha_1 \int_{\Omega} |\nabla \tau_{\mathbf{a}}(\psi)| dx + \frac{\gamma_1}{2} \|v - \tau_{\mathbf{a}}(\psi)\|_{L^2(\Omega)}^2 + \langle \varphi_1, v - \tau_{\mathbf{a}}(\psi) \rangle \\
& \left. + \alpha_2 \int_{\Omega} |\nabla \tau_{\mathbf{b}}(\omega)| dx + \frac{\gamma_2}{2} \|k - \tau_{\mathbf{b}}(\omega)\|_{L^2(\Omega)}^2 + \langle \varphi_2, k - \tau_{\mathbf{b}}(\omega) \rangle \right\}. \quad (7.24)
\end{aligned}$$

In order to solve this model, we derive the partial differential equations defined by the first order optimality conditions. However, the only step that is not identical or closely analogous to the corresponding variable in the previous section is for ψ . We

also briefly present the details of minimising f_{J_2} with respect to the new variable ϖ . First, minimising f_{J_2} with respect to ψ , we obtain the equation

$$\mathcal{E}_2(\psi) = \alpha R(\psi) - \tau_{\mathbf{a}}'(\psi)(\gamma_1 + \varphi_1)(u - \tau_{\mathbf{a}}(\psi)) + \tau_{\mathbf{a}}'(\psi)(v(\varpi - \tau_{\mathbf{a}}(\psi))) - \zeta^\dagger \tau_{\mathbf{a}}(\psi) \quad (7.25)$$

where $R(\psi)$ is the derivative of the regularisation term $\int_{\Omega} |\nabla \tau_{\mathbf{a}}(\psi)| dx$. We can solve $\mathcal{E}_2(\psi) = 0$ using semi-implicit time marching, $\psi_t = -\mathcal{E}_2(\psi)$ by discretising the time step. Now, minimising f_{J_2} with respect to ϖ , we obtain

$$\mathcal{E}_3(\varpi) = 2\lambda(\varpi - c_1)\nu + 2\lambda(\varpi - c_2)(1 - \nu) + u(\varpi - \tau_{\mathbf{a}}(\psi)) + \zeta^\dagger(\varpi - \tau_{\mathbf{a}}(\psi)).$$

Note that we can solve the sub-problem $\mathcal{E}_3(\varpi) = 0$ directly with the solution of

$$\left(2\lambda_1\nu + 2\lambda_2(1 - \nu) + u + \zeta^\dagger\right) \varpi = 2\lambda_1c_1\nu + 2\lambda_2c_2(1 - \nu) + \left(u + \zeta^\dagger\right) \tau_{\mathbf{a}}(\psi). \quad (7.26)$$

In order to solve the model (7.24), we make an initial estimate of the image based on the observed data. We then calculate the initial estimate of ψ as the inverse transform of the initial image. Similarly, in the blind case, we make an initial estimate of the point spread function based on visual observation and compute its inverse transform function. We then proceed to solve the model (7.24), alternately minimising with respect to the arguments. The final segmentation is then given by the contour Γ^* , by thresholding the segmentation function u^* . We present this algorithm in Algorithm 6 below. This is the Relaxed Joint Restoration and Segmentation model, and we refer to it as the RRS model from here.

Algorithm 6 Segmentation of blurred images: $u^{(\ell)} \leftarrow \text{RRS}(u^{(0)}, k^{(0)}, z, \text{maxit})$

- 1: $v^{(0)} \leftarrow z, \varpi^{(0)} \leftarrow v^{(0)}, \psi^{(0)} \leftarrow \xi_{\mathbf{a}}(v^{(0)}), \omega^{(0)} \leftarrow \xi_{\mathbf{b}}(k^{(0)})$
 - 2: $\varphi_1^{(0)} \leftarrow \mathbf{1}, \varphi_2^{(0)} \leftarrow \mathbf{1}, \zeta^{(0)} \leftarrow \mathbf{1}$
 - 3: **for** $\ell \leftarrow 1 : \text{maxit}$ **do**
 - 4: Calculate $c_1^{(\ell)} \leftarrow c_1(\varpi^{(\ell-1)}, \nu^{(\ell-1)}), c_2^{(\ell)} \leftarrow c_2(\varpi^{(\ell-1)}, \nu^{(\ell-1)})$
 - 5: Update $v^{(\ell)}$ and $\psi^{(\ell)}$ using (7.25)
 - 6: Update $k^{(\ell)}$ and $\omega^{(\ell)}$
 - 7: Update $u^{(\ell)}$ and $\varpi^{(\ell)}$ using (7.26)
 - 8: Update $\varphi_1^{(\ell)} \leftarrow \varphi_1^{(\ell-1)} + \gamma_1(u^{(\ell)} - \tau_{\mathbf{a}}(\psi^{(\ell)}))$
 - 9: Update $\varphi_2^{(\ell)} \leftarrow \varphi_2^{(\ell-1)} + \gamma_2(u^{(\ell)} - \tau_{\mathbf{b}}(\omega^{(\ell)}))$
 - 10: Update $\zeta^{(\ell)} \leftarrow \zeta^{(\ell-1)} + v(\tau_{\mathbf{a}}(\psi^{(\ell)}) - w^{(\ell)})$
 - 11: **end for**
-

7.5 Experimental Results

Segmenting a blurred image with a conventional technique (such as Chan-Vese [33]) is sometimes sufficient if the degradation is not strong. However, as the level of corruption

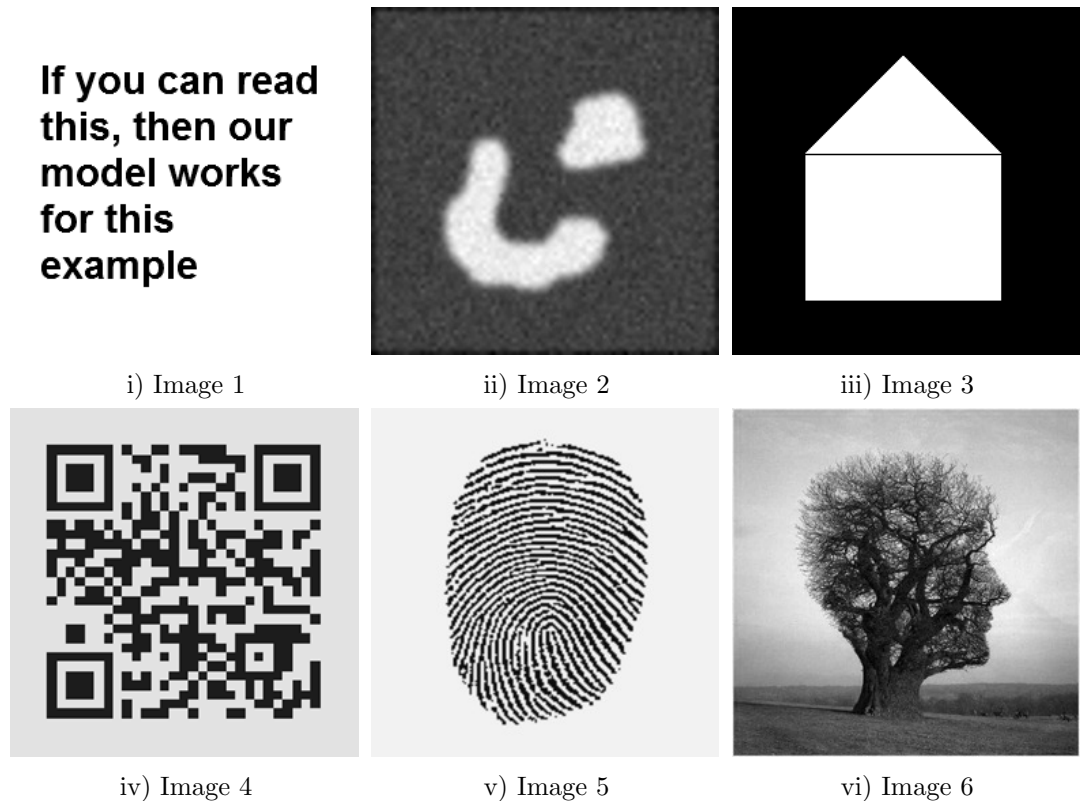


Figure 7.1: Images used for test examples.

increases it is very unlikely to obtain a good result as it is beyond the scope of the formulation. The work of Bar, Sochen and Kiriyati [11], described in Section 7.2, is capable of segmenting blurred images where the corruption is small but is not robust to significant levels of blur degradation or noise.

In this section, we present results of segmenting eight images, and Images 1-6 are shown in Fig. 7.1. In this section, we demonstrate that Algorithms 1 and 2 offer improvements over competitive models for segmenting blurred images. We also show that Algorithm 5 is capable of obtaining an accurate result with the possibility of slow convergence while Algorithm 6 converges faster to a similar, if slightly lower quality result. In order to compare our results with alternative approaches, we define the following models to be tested in this section:

CV: The Chan-Vese segmentation model [33].

GCV: A two-stage model by standard TV deblurring followed by CV segmentation.

PCV: A two-stage model by standard deblurring for Poisson noise followed by CV segmentation.

BSK: The Bar et al. model [11] – without constraints on k_σ, v .

The following are the proposed models we introduced in Sections 7.3 and 7.4:

2SG: Algorithm 3 – Two-Stage Gaussian – implicitly constrained blind deblurring for Gaussian noise followed by GCS (7.12).

2SP: Algorithm 4 – Two-Stage Poisson – Algorithm 3 adapted for Poisson noise.

JRS: Algorithm 5 – Joint Reconstruction and Segmentation model for blind deblurring and GCS with implicit constraints on k, v , from Section 7.3.2.

RRS: Algorithm 6 – Relaxed Reconstruction and Segmentation model from Section 7.4.

In order to quantify the accuracy of our proposed methods we use three measures, that we will now introduce briefly. For artificial images we know the ground truth (GT) for the segmentation result, i.e. the true indicator function of the object. For images that we do not have the ground truth we assume that the segmentation of the true image (i.e. uncorrupted by blur or noise) is correct. As in previous chapters we use the Tanimoto Coefficient [48] to measure the accuracy of the model:

$$TC = \frac{N(\Omega_1^* \cap GT)}{N(\Omega_1^* \cup GT)},$$

where $\Omega_1^* = \{(x) \in \Omega | u^*(x) > 10^{-1}\}$, where $N(\cdot)$ is the number of pixels in the enclosed region. As the restored segmentation approaches the ground truth, TC tends towards one. Additionally, as a measure of the accuracy of the restored segmentation we also measure the L^2 norm of the difference between the computed function, u^* and the ground truth, GT . In other words, this not only measures the accuracy of the thresholded contour, but indicates how close to binary the result is in keeping with the ideas discussed in Chapter 4. We denote this measure $L2A$:

$$L2A = \|u^* - GT\|_2^2.$$

The time taken to compute the restored segmentation is also of interest, particularly in relation to the RRS model. As before, we measure the computation time in seconds and denote it in the following by cpu .

7.5.1 Test Set 1 (Two-Stage Comparisons)

This set of results consists of images corrupted by blur with the assumption that Gaussian noise is present. We illustrate the performance of CV to segment the image and consider it against the performance of GCV and 2SG. We see in Table 7.1 and Figure 7.2 that while CV can give a reasonable result, it is not reliable for segmenting blurred images. We also demonstrate in Table 7.1 that the result can be improved by using the two-stage model, GCV, and further enhanced by the proposed constrained model, 2SG. We also demonstrate in Table 7.2 that this idea carries over to the case of Poisson noise corruption in the image. It can be seen that 2SP offers an improvement over the two-stage Poisson model PCV.

We also consider the comparison between the results obtained by solving the two-stage model, 2SG, with those obtained as the solutions of the joint models JRS and RRS. We can see in Table 7.3 that there is little advantage in considering the problems of deblurring and segmentation separately. In most cases, the joint models achieve better results than the two-stage model.

	Model	$\sigma = 9$		$\sigma = 19$	
		<i>L2A</i>	<i>TC</i>	<i>L2A</i>	<i>TC</i>
Image 1	Initial	101.16	0.13	101.16	0.13
	CV	101.86	0.37	141.48	0.22
	GCV	15.32	0.96	44.82	0.85
	2SG	13.09	0.97	17.24	0.95
Image 2	Initial	104.31	0.23	104.31	0.23
	CV	32.27	0.89	63.59	0.67
	GCV	18.99	0.97	14.73	0.89
	2SG	7.27	0.99	10.50	0.99
Image 3	Initial	109.65	0.42	109.65	0.42
	CV	16.79	0.99	32.10	0.95
	GCV	1.62	1.00	1.71	1.00
	2SG	1.57	1.00	1.56	1.00
Image 4	Initial	138.19	0.19	138.19	0.19
	CV	81.87	0.74	142.16	0.45
	GCV	21.69	0.89	48.71	0.88
	2SG	17.94	0.98	20.29	0.98
Image 5	Initial	109.07	0.29	109.07	0.29
	CV	120.87	0.49	134.30	0.42
	GCV	44.96	0.87	46.33	0.86
	2SG	16.11	0.98	27.18	0.95
Image 6	Initial	145.97	0.27	145.97	0.27
	CV	69.01	0.85	84.28	0.79
	GCV	46.22	0.88	35.91	0.95
	2SG	18.01	0.99	27.74	0.97

Table 7.1: Test Set 1. Error values for Images 1-6 corrupted by Gaussian blur and segmented by CV. In many cases, the competition is close but 2SG obtains the same or improved error values over competing models in all cases.

7.5.2 Test Set 2 (Significant Blur)

These results consist of images corrupted by small and large amounts of blur and noise. We demonstrate in Table 7.6 and Figures 7.2–7.5 that CV is sometimes sufficient to obtain a fairly close result but misses a considerable amount of detail while BSK [11] can give an improvement over this. We also demonstrate that the new joint models JRS and RRS are capable of segmenting these examples and offer further enhancement over BSK. Further to this, we demonstrate in Table 7.4 and Figure 7.3 that, when the level of blur is larger, JRS offers further improvements over BSK. Finally, we demonstrate in Table 7.5 and Figure 7.4 that, as the level of noise is increased, JRS continues to perform well.

7.5.3 Test Set 3 (Joint Model Comparisons)

Test Set 4 demonstrates the ability of RRS and compares the performance of this model with JRS. We can see in Tables 7.5–7.6 and Figure 7.7 that it is generally the case that RRS is faster than JRS while JRS obtains better results.

	Model	$\sigma = 9$		$\sigma = 19$	
		<i>L2A</i>	<i>TC</i>	<i>L2A</i>	<i>TC</i>
Image 1	CV	101.86	0.37	141.48	0.22
	PCV	72.97	0.54	96.44	0.39
	2SP	67.52	0.57	90.31	0.43
Image 2	CV	32.27	0.89	63.59	0.67
	PCV	11.74	0.98	25.72	0.92
	2SP	11.74	0.98	25.50	0.92
Image 3	CV	16.79	0.99	32.10	0.95
	PCV	12.29	0.99	12.49	0.99
	2SP	11.97	0.99	12.28	0.99
Image 4	CV	81.87	0.74	142.16	0.45
	PCV	57.22	0.86	110.57	0.61
	2SP	55.51	0.86	110.51	0.61
Image 5	CV	120.87	0.49	134.30	0.42
	PCV	102.98	0.57	104.25	0.56
	2SP	102.95	0.57	104.21	0.56
Image 6	CV	69.01	0.85	84.29	0.79
	PCV	50.96	0.91	57.27	0.89
	2SP	50.96	0.91	57.27	0.89

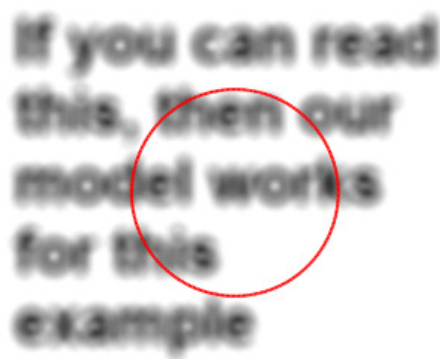
Table 7.2: Test Set 1. Error values for Images 1-6 corrupted by Gaussian blur and segmented by PCV and 2SP. The competition is close for most examples, but overall 2SP outperforms PCV.

	Initial	2SG	JRS	RRS
Image 1	101.16	13.09	15.34	16.59
Image 2	104.31	7.27	5.82	6.92
Image 3	109.65	1.57	1.09	1.52
Image 4	138.19	17.94	14.36	17.63

Table 7.3: Test Set 1. Error values given by *L2A* for Images 1-4 corrupted by Gaussian blur and segmented by 2SG, JRS and RRS. For Image 1, 2SG outperforms the other models but in the remaining cases JRS and RRS obtain improved results.

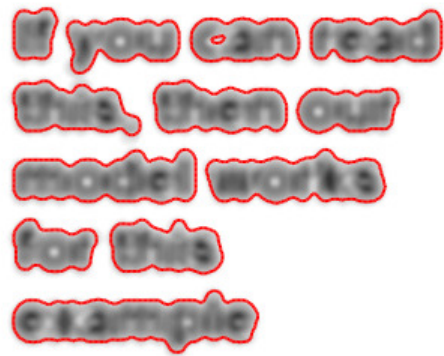
7.6 Remarks

In this chapter we have proposed a new model for the joint reconstruction and segmentation of blurred images where the blur function is unknown, which we call blind image segmentation (JRS model). The results presented in Section 7.5 demonstrate strong performance for images where the edges are unclear visually, and beyond conventional segmentation methods discussed in Chapters 5 and 6, and in [33, 30, 18, 37]. We have also presented an accelerated model (RRS model) which is capable of achieving results of a similar standard to JRS. Crucially, both joint models offer an improvement over analogous two-stage methods (2SG and 2SP) demonstrating the effectiveness of this formulation over alternative methods [97], and another model from Bar et al. [11] (BSK) that was tested here. The proposed model can be extended to the semi-blind case [4, 12, 86, 96, 99, 110] which can offer speed improvements, and examples involv-



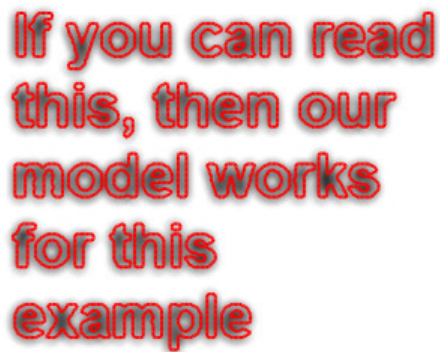
If you can read
this, then our
model works
for this
example

i) Initial Contour




If you can read
this, then our
model works
for this
example

ii) CV, Γ^*



If you can read
this, then our
model works
for this
example

iii) JRS, Γ^*



If you can read
this, then our
model works
for this
example

iv) JRS, u^*

Figure 7.2: Test Sets 1 and 2. Illustration of the performance of CV for Image 1 corrupted by Gaussian blur: i) Initial contour. ii) Segmentation given by CV. iii)-iv) segmentation given by JRS. CV gives a rough segmentation while the spaces between the letters which are hidden by the blur are successfully segmented using JRS.

ing multi-channel images. For the purposes of speeding up the segmentation step, it is possible to use the improved AOS schemes introduced in Chapter 4, or apply methods such as Split Bregman [58] and Chambolle-Pock [26].

Model	<i>L2A</i>	<i>TC</i>	<i>cpu</i>	<i>L2A</i>	<i>TC</i>	<i>cpu</i>
	Image 1			Image 5		
Initial	101.16	0.13		109.07	0.29	
CV	141.48	0.22	520	134.30	0.42	523
BSK	151.08	0.21	1479	126.19	0.46	937
JRS	35.68	0.80	557	74.10	0.71	564
RRS	44.93	0.75	549	74.77	0.69	334
	Image 2			Image 6		
Initial	104.31	0.23		145.97	0.27	
CV	63.59	0.67	522	84.29	0.79	520
BSK	47.17	0.79	950	76.36	0.82	940
JRS	6.79	0.99	566	40.19	0.94	574
RRS	10.17	0.99	434	42.54	0.94	539
	Image 3			Image 7		
Initial	109.65	0.42		65.21	0.69	
CV	32.10	0.95	418	114.90	0.47	525
BSK	28.31	0.96	730	124.12	0.44	1364
JRS	11.96	0.99	552	26.29	0.93	543
RRS	12.08	0.99	232	24.81	0.95	430
	Image 4			Image 8		
Initial	138.19	0.19		120.54	0.32	
CV	142.16	0.45	525	89.11	0.70	522
BSK	140.11	0.49	1477	87.84	0.72	1148
JRS	25.03	0.96	589	47.29	0.89	548
RRS	30.42	0.95	444	43.29	0.90	539

Table 7.4: Test Set 2. Error values and *cpu* times (in seconds) for images Images 1-8 corrupted by strong Gaussian blur. In all cases, JRS and RRS achieve improved results and competition is close between JRS and RRS. For most cases, the *cpu* time is lower for RRS with the exception of three examples which have slightly lower *cpu* time for CV with deteriorated results.

Model	<i>L2A</i>	<i>TC</i>	<i>cpu</i>	<i>L2A</i>	<i>TC</i>	<i>cpu</i>
	Image 1			Image 4		
Initial	101.16	0.13		138.19	0.19	
CV	101.84	0.37	319	81.91	0.74	530
BSK	94.93	0.40	1899	65.71	0.82	958
JRS	13.86	0.97	607	26.84	0.96	640
RRS	14.90	0.96	452	23.90	0.97	342
	Image 3			Image 5		
Initial	109.65	0.42		109.07	0.29	
CV	16.86	0.99	317	120.86	0.49	547
BSK	14.32	0.99	633	113.25	0.52	877
JRS	1.75	1	512	41.61	0.87	600
RRS	2.18	1	242	45.32	0.86	575

Table 7.5: Test Set 2. Error values and *cpu* times (in seconds) for Images 1, 3-5 corrupted by Gaussian blur and noise. In all cases, JRS and RRS achieve improved results. *cpu* time is lower for RRS in two cases. In the remaining cases, it is lower for CV and closely followed by RRS which achieved significantly improved results.

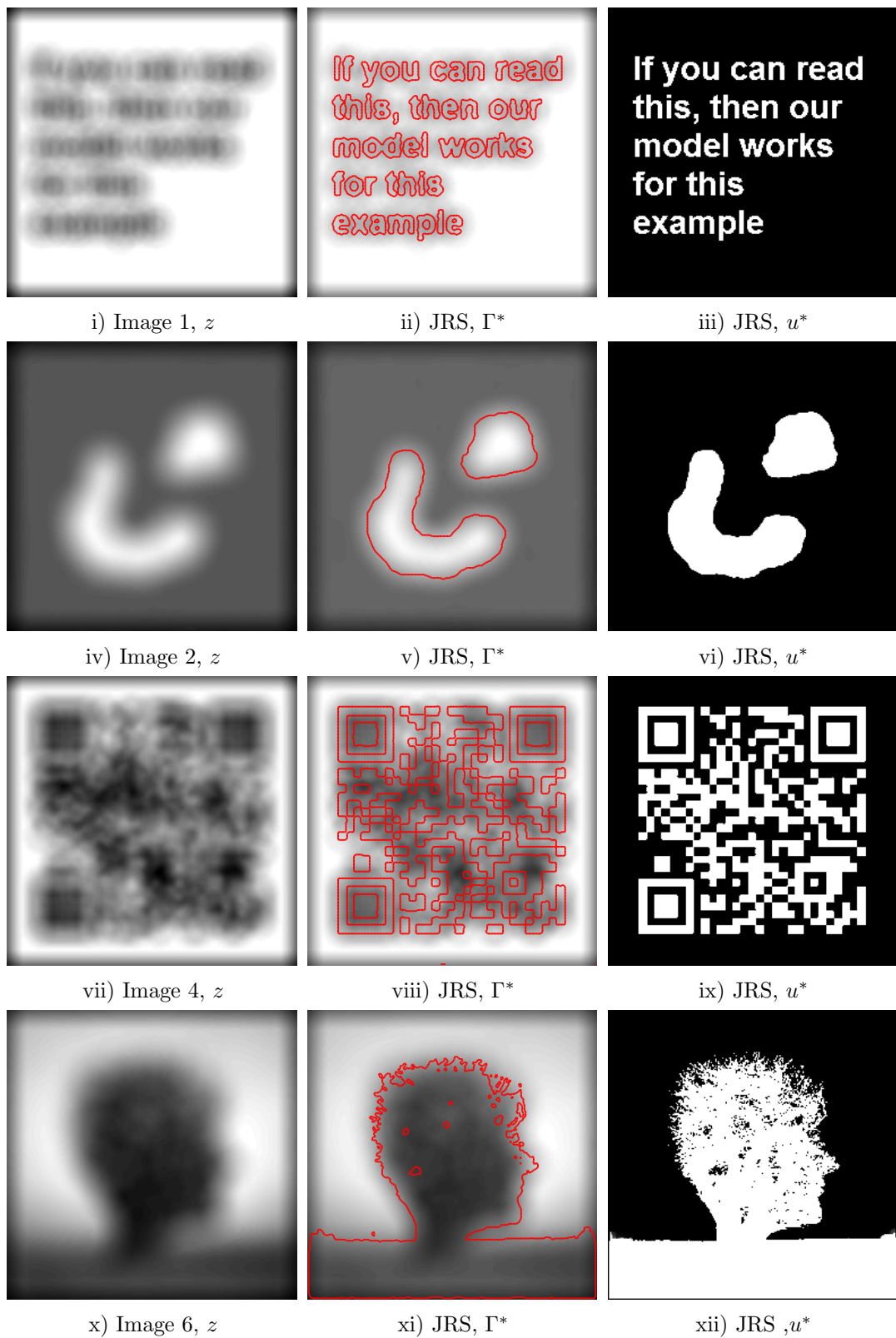


Figure 7.3: Test Set 2. Illustration of the performance of the JRS for (top-bottom) Image 1, 2, 4, and 6 corrupted by strong Gaussian blur. JRS is capable of segmenting edges in these challenging cases which cannot be segmented by CV.

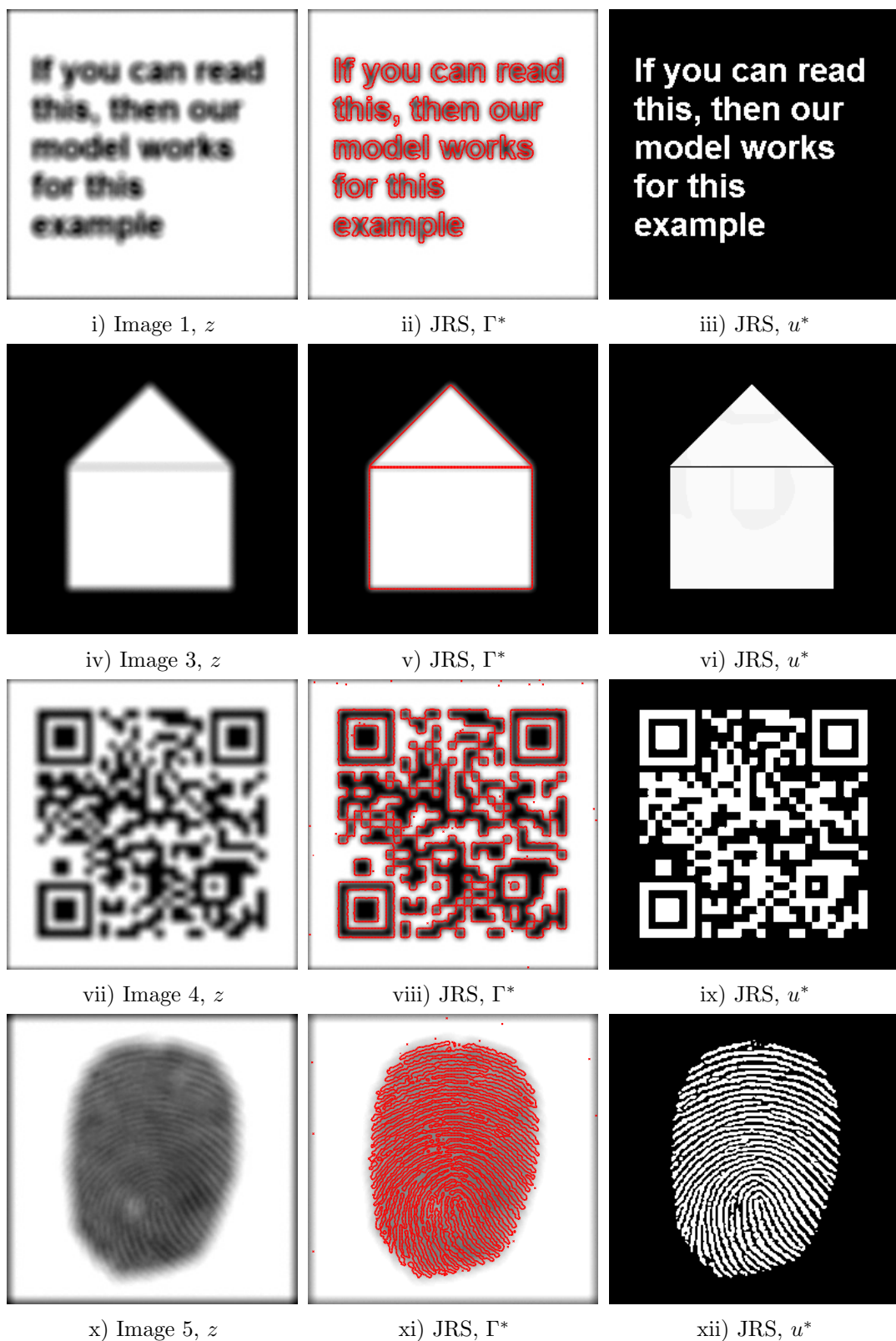


Figure 7.4: Test Set 2. Illustration of the performance of the JRS for (top-bottom) Image 1,3,4, and 5 corrupted by Gaussian blur and noise. The edges hidden by blur are successfully segmented by JRS which cannot be segmented by CV.

Model	<i>L2A</i>	<i>TC</i>	<i>cpu</i>	<i>L2A</i>	<i>TC</i>	<i>cpu</i>
	Image 1			Image 5		
Initial	101.16	0.13		109.07	0.29	
CV	101.86	0.37	317	120.87	0.49	524
BSK	94.94	0.40	1362	113.26	0.52	835
JRS	15.34	0.96	559	28.43	0.93	568
RRS	16.59	0.95	443	38.05	0.90	543
	Image 2			Image 6		
Initial	104.31	0.23		145.97	0.27	
CV	32.27	0.89	527	69.01	0.85	525
BSK	23.02	0.94	861	61.13	0.88	836
JRS	5.82	0.99	546	32.41	0.96	562
RRS	6.92	0.99	427	35.34	0.95	537
	Image 3			Image 7		
Initial	109.65	0.42		65.21	0.69	
CV	16.79	0.99	319	37.52	0.89	423
BSK	13.64	0.99	634	33.10	0.91	731
JRS	1.09	1	550	18.63	0.97	330
RRS	1.52	1	228	18.59	0.97	320
	Image 4			Image 8		
Initial	138.19	0.19		120.54	0.32	
CV	81.87	0.74	527	78.07	0.76	538
BSK	65.75	0.82	942	72.55	0.78	1166
JRS	14.36	0.99	586	34.13	0.93	561
RRS	17.63	0.98	331	32.36	0.94	542

Table 7.6: Test Sets 2 and 3. Error values and *cpu* times (in seconds) for images Images 1-8 corrupted by small Gaussian blur. In all cases, JRS and RRS achieve improved results with JRS typically achieving better results. For many examples, the *cpu* time is lower for CV but it is closely followed by RRS which gives considerably better results.

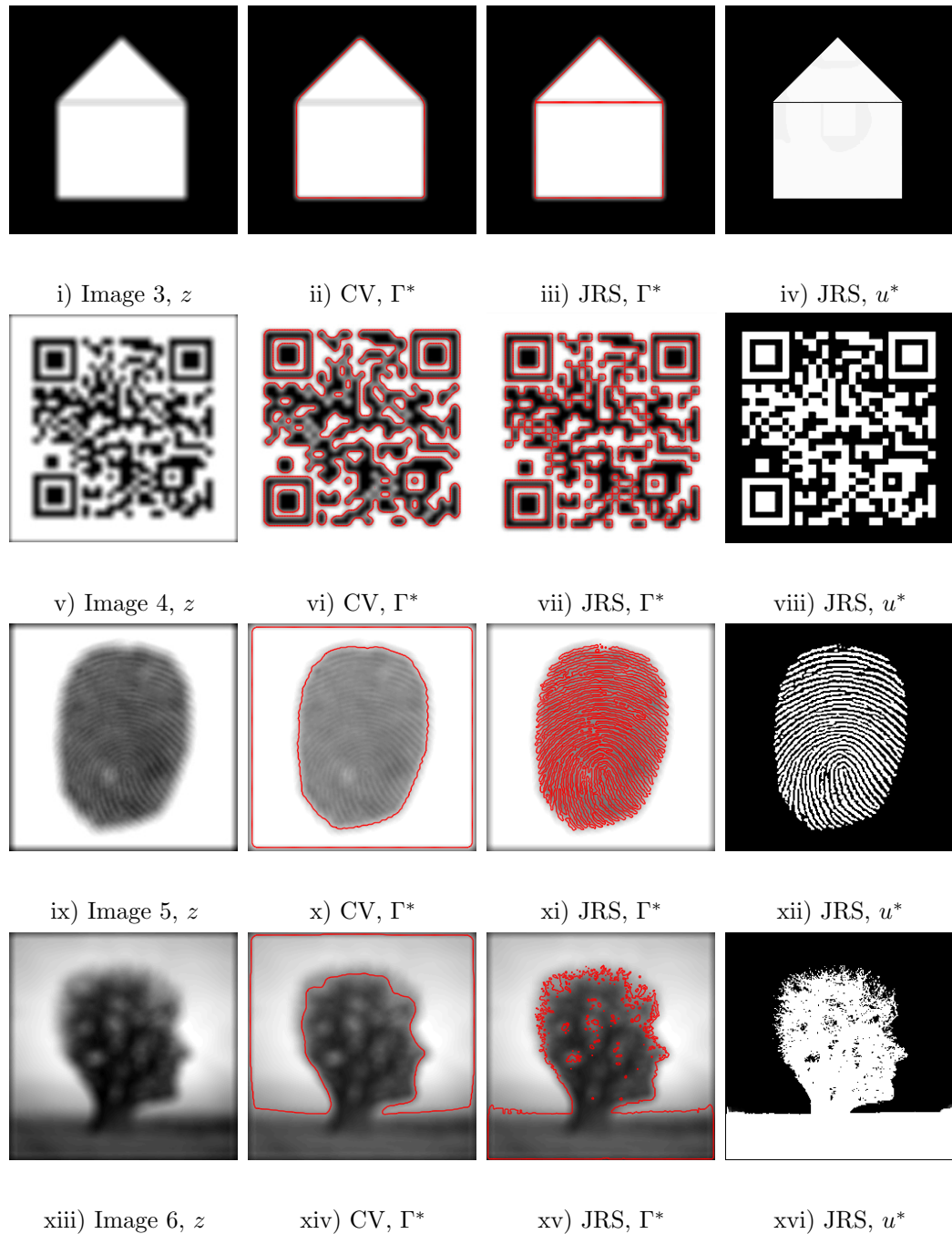


Figure 7.5: Test Sets 2 and 3. Illustration of the performance of JRS for (top-bottom) Image 3,4,5, and 6 corrupted by Gaussian blur. The edges hidden by blur are successfully segmented by JRS which cannot be segmented by CV.

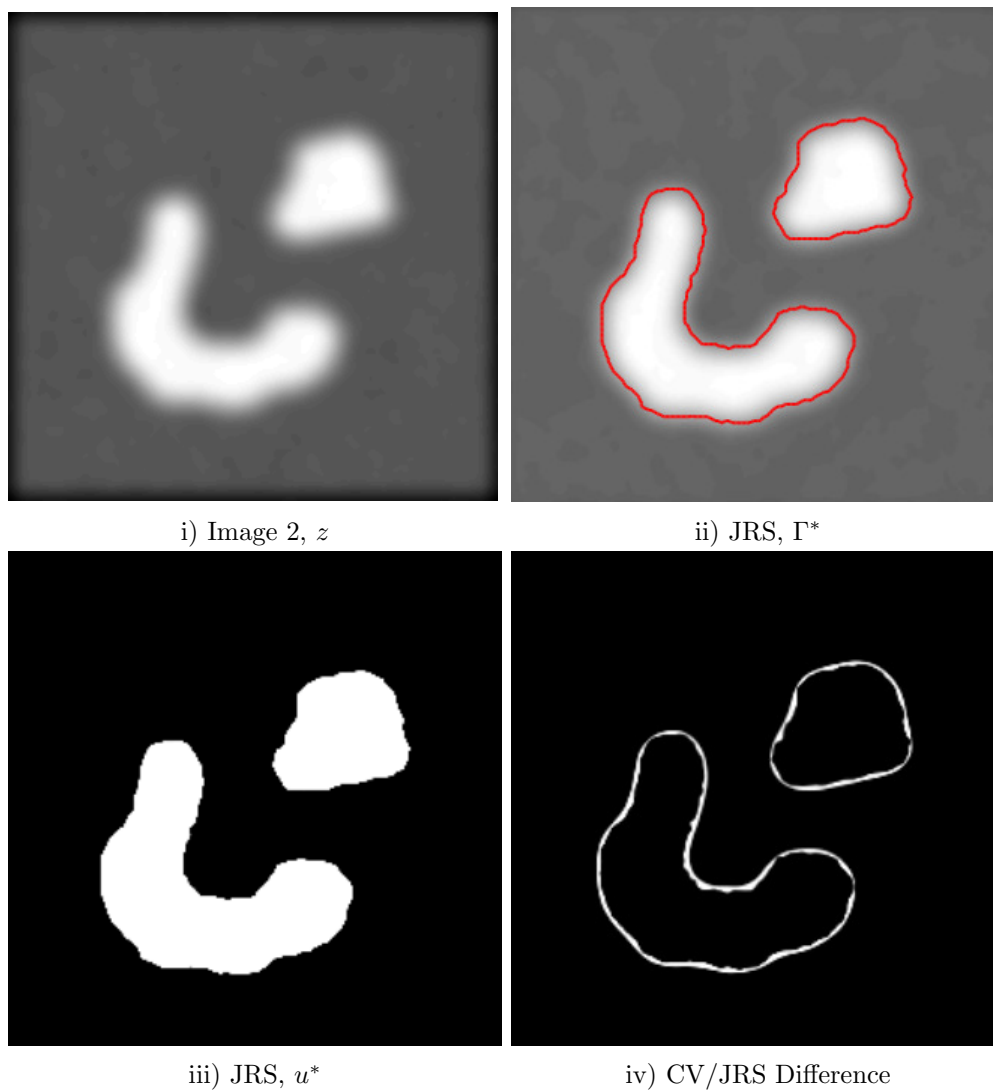


Figure 7.6: Test Sets 2 and 3. Illustration of the performance of the JRS for Image 2 corrupted by Gaussian blur: i) Received data. ii)-iii) Segmentation using JRS. iv) the difference between the segmentation using JRS and using CV. The segmentation is closer to the true edge using JRS while CV also captures the blurred edge.

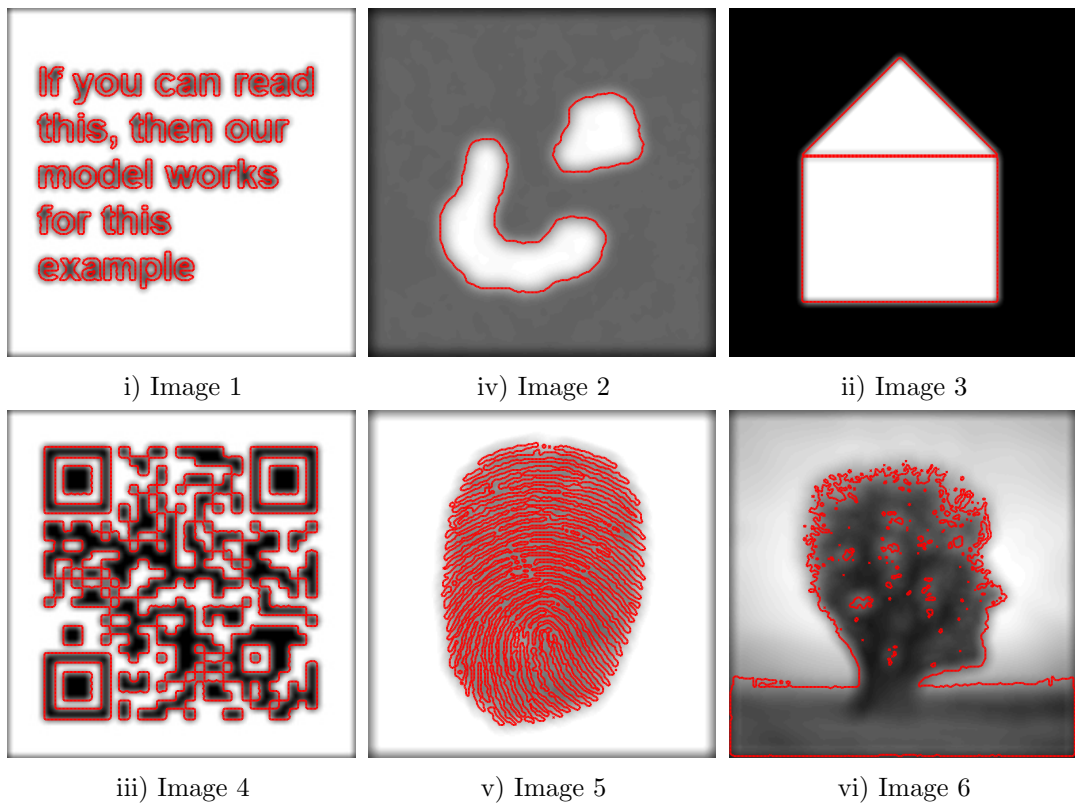


Figure 7.7: Test Set 3. Images corrupted by Gaussian blur segmented using RRS.

Chapter 8

Incorporating Shape Priors in Variational Segmentation

8.1 Introduction

In Chapter 4 we introduced a general approach to compute global minimisers of GCS models, with a generalised fitting function. In previous chapters we have mainly discussed two-phase segmentation problems where the foreground and background are primarily distinguished based on the intensity of the observed data. In practice, information about the target object is often known a priori and incorporating this information into a model has clear advantages. This enables us to increase the robustness of our approach in terms of accuracy, speed, and reliability, and it can also improve the quality of results in the case of challenging data that are beyond the scope of conventional intensity based approaches [89, 33, 37]. In this chapter, we address the task of identifying objects in an image based on their similarity to a shape prior, i.e. we formulate the fitting function in GCS based on a shape prior.

Previous approaches have incorporated shape information into variational segmentation models, such as the seminal work of Cremers et al. [46] known as 'Diffusion Snakes' where statistical shape information was combined with the piecewise-constant formulation of Chan-Vese [33]. Other important early work includes Leventon et al. [80], which was based on using shape priors with the Geodesic Active Contours model of Caselles et al. [22]. Closely related is the approach of Chen et al. [40] and Rousson and Paragios [108], both introduced in 2002.

The work of particular importance to our approach is that of Cremers et al. [47], who introduced a labelling function to indicate which regions in an image the shape prior should be enforced. Based on this work, Chan and Zhu [36] introduced a similar method where the prior was permitted to be scaled, rotated, and translated, increasing the effectiveness of the approach. Here, the shape is represented as a signed distance function and the segmentation is in the level set framework similar to [33, 105]. A more recent approach is that of Pock et al. [130], who use a convex formulation of Geodesic Active Contours [22], similar to Bresson et al. [18]. The shape prior is also defined as a distance function that is again permitted to be scaled, rotated, and translated.

This chapter is organised as follows. In Section 2, we will discuss the background

behind incorporating shape information into a segmentation model. Particularly, we will provide some details about the registration step where the shape prior is transformed to match the observed data. We will also briefly review two important models mentioned above: Chan and Zhu [36] and Pock et al. [130]. Additionally, we will highlight how each model represents shape and establish a basis for a new approach. In Section 3, we will discuss the motivation behind our method where the shape can be approximated based on intensity fitting terms and the challenges associated with this idea. In Section 4, we propose a two-stage shape prior model and detail the registration and segmentation steps. Finally, in Section 5, we present results for a range of examples and make some concluding remarks in Section 6.

8.2 Background and Related Models

Incorporating a shape prior into a segmentation model involves transforming a given shape prior to fit the observed data, which involves image registration. Typically, this involves parametric methods where the transformation is rigid or affine [36, 130]. This is because it ensures the final segmentation result is closely related to the prior information. Alternative registration methods that allow deformations are non-parametric methods such as [66, 138] and some segmentation methods incorporate similar nonlinear registration steps, such as Cremers et al. [39].

We now briefly discuss parametric registration methods, also discussed in Chapter 3. The template and reference, $T, R \in \Omega$, define the images to be compared. For $x \in \Omega$ denote by $\phi(x) : \Omega \rightarrow \Omega$ the unknown coordinate transformation that produces the alignment between the reference, $R(x)$ and the transformed template, $T(\phi(x))$. We address the problem where we assume the target object is approximately an affine transformation of the shape prior, such that the segmentation closely favours shapes given by the prior information. This means the transformation is linear and can be defined as follows:

$$\phi(x) = \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \end{bmatrix}, \quad (8.1)$$

given six parameters $\mathbf{a} = \{a_1, a_2, a_3, a_4, a_5, a_6\}$. The regularised affine registration model [41] is as follows:

$$\min_{\mathbf{a}} \left\{ \eta \mathcal{R}(\mathbf{a}) + \frac{1}{2} \int_{\Omega} (T(\phi) - R(x))^2 dx \right\},$$

where $\eta > 0$ is a weighting parameter for the regularisation of \mathbf{a} , and the sum of squared differences (SSD) term determines the similarity between the reference and template. In the following sections we detail how linear registration methods are incorporated into variational segmentation models.

8.2.1 Level Set Based Shape Prior Segmentation

In 2005, Chan and Zhu [36] introduced a shape prior model in the level set framework. It is based on the work of Cremers et al. [47], who introduced a dynamic labelling function, L , that automatically indicates which region the shape prior should be applied. The shape prior is in the form of a distance function, similar to the level set function φ used in many segmentation methods [33, 112]. This is a natural choice due to the framework used for the segmentation step. The following are relevant terms from the Chan and Zhu functional:

$$\int_{\Omega} (H(\varphi)H(L) - H(\psi))^2 dx + \int_{\Omega} \left[\lambda_1 f(x)H(\varphi) + \lambda_2 f(x)H(\psi) \right] dx,$$

where $f(x) = (z - c_1)^2 - (z - c_2)^2$, assuming fixed c_1 and c_2 . Additional terms provide regularisation for L , and details of how the shape prior is translated, scaled, and rotated can be found in [36]. These terms are interesting as they demonstrate the approach for this model. The shape matching and intensity fitting are treated separately, with parameters providing the balance between the two. The approach of Chan and Zhu demonstrates positive results, especially in the case of occlusions, which are common in real images and prevent an accurate segmentation being found. However, they discuss its dependence on the initialisation as being a drawback.

8.2.2 Interactive Shape Prior Segmentation

In 2009, Pock et al. [130] introduced a semi-automated method based on the Geodesic Active Contours energy [22, 18] incorporating a shape prior. The shape prior is given by $\psi(x)$, which must be positive outside the shape and negative inside the shape. In Cremers et al. [47], they use a binary function with $\psi(x) = 1$ and $\psi(x) = -1$ outside and inside the shape respectively. Pock et al. [130] discuss the drawbacks of representing a shape with a binary function, and instead opt to use a distance function to implicitly represent the shape prior. They introduce parameters to define the translation, scaling, and rotation of the prior, such that the prior is given by $\psi(\phi)$ (n.b. affine registration can be restricted to scaling, rotation, and translation easily. We use the notation of (8.1) for convenience). The joint minimisation problem is given as:

$$\min_{u, \mathbf{a}} \left\{ \int_{\Omega} g(x) |\nabla u(x)| dx + \lambda \int_{\Omega} \psi(\phi) u(x) dx \right\} \quad (8.2)$$

Pock et al. [130] approached the above problem (8.2) with a semi-automated alternate minimisation scheme. The first step was to complete the segmentation step, with fixed transformation parameters, in which they use a primal-dual formulation. For the shape alignment stage it is possible to do a complete search over the whole parameters space of \mathbf{a} , as discussed by Cremers et al. [45]. However, due to time restrictions they opt to restrict the domain of \mathbf{a} such that the minimisation can be achieved in real time. The user can position the shape in a suitable way, and gets an opportunity to interact with the algorithm. The optimal parameters are determined by the position the energy attains its minimum within this domain. This procedure is then iterated

until convergence. Based on the results presented, this method is capable of achieving impressive results for a wide range of examples.

8.2.3 Shape Representation

Both of the previous models discussed, as well as many others [80, 108, 40, 47], represent a given shape implicitly by a distance function (see Fig. 8.1). This term is incorporated into the functional in different ways in [36, 130] (i.e. Chan-Vese fitting and edge detection) but the premise is similar in the sense that the shape and intensity terms are distinct and are balanced by a weighting parameter. We have observed two problems with this approach. First, the representation of a given shape with a level set function is not unique so if similar shapes are defined in an inconsistent manner aligning them can be problematic. Secondly, we have found that constructing a prior in this way can create a sensitive parameter dependence, which we will discuss further in Section 8.5.2. Our method combines the intensity information in the observed data and the shape prior in a reliable way, and we discuss the motivation behind this idea in the next section.

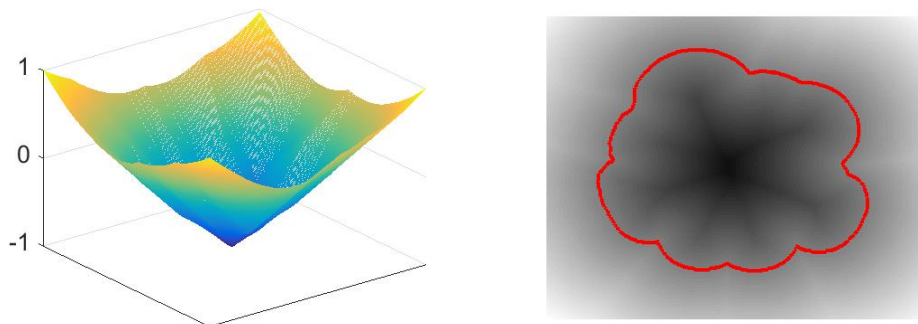


Figure 8.1: Chan and Zhu [36] and Pock et al. [130] represent shapes as distance functions, $\psi(x)$. The shape is implicitly defined as the zero level set of $\psi(x)$ (given in red).

8.3 Motivation

In this section, we discuss the main ideas considered in formulating our shape prior approach. The three main considerations from Section 8.2 are as follows, based on previous approaches to this problem such as Chan and Zhu [36] and Pock et al. [130]. Firstly, what is the most appropriate representation for a given shape? Secondly, how should the shape prior term interact with additional fitting terms? Thirdly, what is the most effective method to discriminate between similar shapes in an image?

We'll start by recalling what constitutes a successful fitting term for a simple two-phase image that is approximately piecewise-constant. Fig. 8.2 demonstrates the following fitting function:

$$f(z) = (z - c_1)^2 - (z - c_2)^2, \quad (8.3)$$

where c_1 and c_2 are the known constants of average intensity inside and outside the object respectively. If an image $z(x)$ can be approximated by a piecewise-constant function then the fitting term $f(z)$ will closely describe the boundary of the object. When included in the following convex relaxation framework, we can observe important characteristics of this fitting function:

$$\min_u \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(z)u(x) dx \right\}$$

for $u \in BV(\Omega; [0, 1])$ and Γ is the boundary of $\Sigma(\gamma) = \{x : u(x) > \gamma\}$, as defined in Chapter 4. This is based on the work of Chan, Esedoglu, and Nikolova [30], Bresson et al. [18], and many others as discussed in previous chapters [102, 79, 25]. Here, where $f(z)$ is negative it is likely that $u(x) = 1$, and where $f(x)$ is positive $u(x) = 0$. The regularisation term, $TV(u)$, penalises the length of the contour Γ and is balanced by the fitting parameter $\lambda > 0$. For a small λ , Γ will be smooth, and as λ increases Γ will resemble the zero level set of $f(z)$.

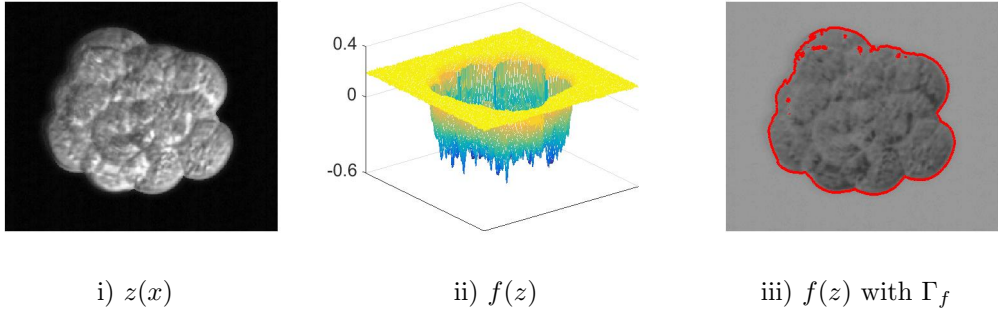


Figure 8.2: The fitting term, $f(z)$, for a given image, z , based on known intensity constants c_1 and c_2 . Here, $\Gamma_f = \{x : f(z) = 0\}$ is given in red in iii) and approximates the shape of the object in z .

With this in mind, we can formulate a shape prior using an approximation of the shape from the fitting function. We assume that our prior information consists of an image, $z_p(x)$, and its ground truth segmentation, which we denote $u_p(x)$. With this an optimal choice of c_1 and c_2 can be computed and used in the fitting term (8.3). In an affine registration framework there is a reference and a template. In this context the template should be a shape prior that is transformed based on the parameters \mathbf{a} (8.1), fitting a reference that corresponds to the observed image data, z . We define the shape prior function, which acts as a template in the registration step:

$$S(x) = -H(-f(z_p))f(z_p). \quad (8.4)$$

The reference is based on the observed data and is constructed in a similar way:

$$F(z) = -H(-f(z))f(z), \quad (8.5)$$

where H is the Heaviside function as defined in Chapter 2. An example of the function $S(x)$ can be seen in Fig. 8.3. The restrictions on this term are that the prior data, z_p , should be similar to the observed data, z . Specifically, their similarity should be in the sense that their fitting functions f should be comparable such that an affine registration between the two is feasible.

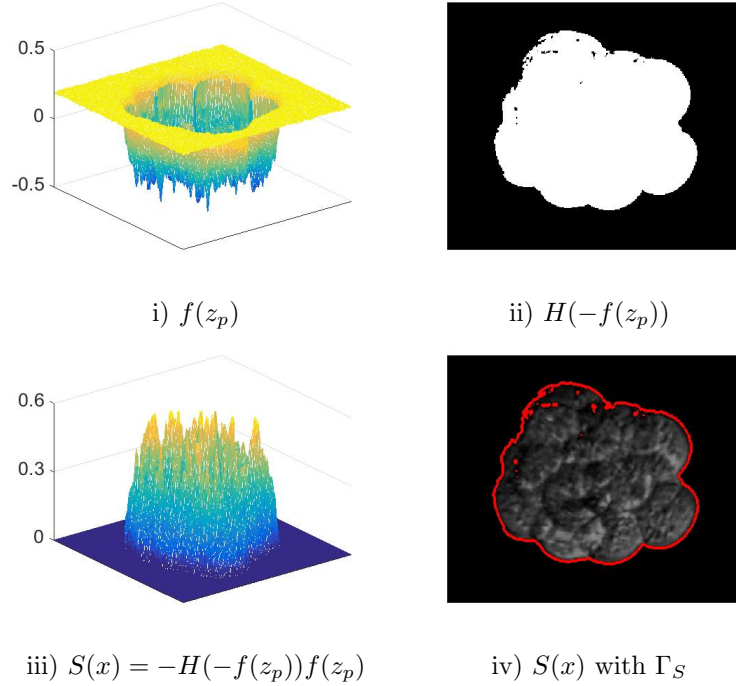


Figure 8.3: The shape prior term, $S(x)$, based on the prior image z_p . Here, $\Gamma_S = \{x : S(x) > \omega\}$ is given in red (for small ω) in iv) and approximates the shape of the object in z_p . The template is formed in this way such that $S(x) \in [0, 1]$.

This gives us a new shape representation for a prior, $S(x)$, that is based on similarities between the observed and prior data. This helps avoid some of the problems discussed in Section 8.2 with respect to shape representation. When transformed by the parameters \mathbf{a} , $S(x)$ should match the fitting function of the observed data. This gives us a registration step in our proposed model:

$$\min_{\mathbf{a}} \left\{ \eta \mathcal{R}(\mathbf{a}) + \frac{1}{2} \int_{\Omega} \left(S(\phi) - F(z) \right)^2 dx \right\},$$

where $\mathcal{R}(\mathbf{a})$ is a regularisation term. The shape term, $S(\phi)$, defines the fitting term for the interior of the object and the fitting term for the exterior of the object can be constructed from the observed data, to form a complete fitting term:

$$\alpha H(f(z))f(z) - S(\phi), \quad (8.6)$$

where α is a positive weighting parameter between the interior and exterior fitting. In this way the shape term and the intensity fitting terms are intertwined, in contrast to the framework of Chan and Zhu [36], and Pock et al. [130]. This just leaves the consideration of how to distinguish between similar shapes in the observed data. As discussed

in the previous section, Chan and Zhu use a dynamic labelling function L , where as Pock et al. use semi-automated positioning, and interaction with the algorithm. We propose a proximity function $P(u)$ that is incorporated into the registration term. It should be analagous to the labelling function of Chan and Zhu in that it indicates where the shape fitting should take place. I will not discuss the specifics of this term at this stage, for reasons that will become clear, but will return to it later. With the above considerations in mind the joint formulation we initially propose is as follows:

$$E(u, \mathbf{a}) = \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} [\alpha H(f)f - S(\phi)] u(x) dx + \eta \mathcal{R}(\mathbf{a}) + \frac{1}{2} \int_{\Omega} (S(\phi) - P(u)F(z))^2 dx. \quad (8.7)$$

The idea is to minimise this functional alternately for the affine parameters \mathbf{a} and the segmentation function u :

$$\min_{u, \mathbf{a}} E(u, \mathbf{a}).$$

The problem with minimising the above functional with respect to u and \mathbf{a} are that both the registration and segmentation steps are complicated by the proposed terms and the way they interact. For the registration step, there are contradictory fitting terms with one acting in a restricted domain. For the segmentation step, the convex relaxation framework is contradicted and the choice of $P(u)$ must satisfy additional conditions which makes its selection difficult. However, it is possible to retain the ideas behind this formulation whilst simplifying the minimisation. This leads us to the idea of splitting the functional into a two-stage model. We describe this in the following section.

8.4 Proposed Two-Stage Shape Prior Model

Splitting the process into two stages simplifies it significantly, and we have found that we can retain the advantages discussed in the previous section whilst doing so. The first stage concerns the affine registration of the interior fitting terms, and the second stage deals with the segmentation. By separating each step in this way it simplifies the registration stage, such that a result can be reliably achieved with conventional methods quickly. Also, we want to retain the convex relaxation framework due to its reliability in finding the global minimum independently of initialisation [18, 30, 112]. To achieve this we choose to fix $P(u)$ in the SSD term, such that it is not involved in the minimisation of u but still influences the registration stage. With this adjustment, the choice of this proximity function is much more straightforward and is dependent on a fixed function related to the binary prior u_p . We define a translation of the binary prior, that we call \tilde{u} , that is based on user input $\tilde{\mathbf{a}} = \{1, 0, \tilde{a}_3, 0, 1, \tilde{a}_6\}$. The idea is for the user to position u_p such that it is centred on the target object in z :

$$\tilde{u}(x) = u_p(\tilde{\phi}), \quad \text{where } \tilde{\phi} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \tilde{a}_3 \\ \tilde{a}_6 \end{bmatrix}. \quad (8.8)$$

The selection of $\tilde{\mathbf{a}}$ is important when the location of the prior, u_p , is not close to the target object in the domain and when there are multiple objects in z , particularly of a similar shape. This kind of interaction is analogous to the alignment step of Pock et al. [130], as discussed in Section 8.2.2. However, it is simplified in the sense that it is a single act that does not need periodic corrections to ensure an accurate result. It is worth noting here that there are many possible alternatives to incorporating simple user interaction to influence the location of the template; examples include landmark based registration methods such as Lui et al. [71]. It is also possible to use centroid constraints as discussed in Klodt and Cremers [75], which would be applied at the segmentation stage.

The principle behind the proximity function, $P(u)$, is that it should favourably weight the fitting term, $F(z)$, close to where $\tilde{u} = 1$. An obvious choice is

$$P_\beta(\tilde{u}) = 1 - \frac{1}{\beta} \min\{\beta, d(\tilde{u})\}, \quad (8.9)$$

where $d(\tilde{u})$ is the normalised Euclidean distance from the translated object prior, \tilde{u} . The parameter $\beta > 0$ has been included to make the influence of the fitting term, $F(z)$, vary based on the intensity of the image. In images that contain a single object, $\beta = 1$ is appropriate. For more difficult examples, a smaller value of β is necessary and is based on the distance between objects. The function, $P_\beta(\tilde{u})$, is defined in this way to impose some consistency on the function such that its influence is easier to predict and parameter selection is more reasonable. Some examples of what this function looks like for different choices of β is shown in Fig. 8.4.

It can then be seen from the SSD term in the joint formulation (8.7) that with this choice of proximity function, $P_\beta(\tilde{u})F(z)$ has the effect of approximating the fitting term of the target object. This gives us a new SSD term for our proposed model:

$$\frac{1}{2} \int_{\Omega} \left(S(\phi) - P_\beta(\tilde{u})F(z) \right)^2 dx.$$

We denote the result of minimising this functional as $S(\phi^*)$, which we will define precisely in Section 8.4.1. As before, the background fitting can be constructed, again weighted by a parameter $\alpha > 0$. Our proposed combined fitting term is:

$$h(x) = \alpha H(f)f - S(\phi^*),$$

In the following we discuss the details of our two-stage algorithm, beginning with the affine registration step.

8.4.1 Stage 1: Affine Registration

First, let's summarise the registration step, where a template based on the fitting function of a prior image is transformed to match a reference given by a localised

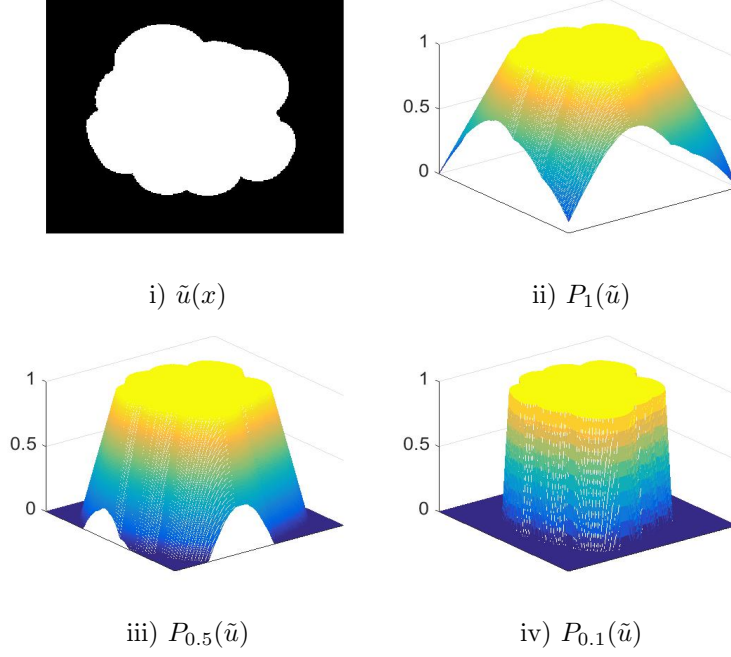


Figure 8.4: The choice of proximity function: $P_\beta = 1 - \frac{1}{\beta} \min\{\beta, d(\tilde{u})\}$, where $d(\tilde{u})$ is the normalised Euclidean distance from the object prior. This forms the reference for Stage 1 (Affine registration), $P_\beta(\tilde{u})F(z)$. i) The translated binary prior, \tilde{u} , given by (8.8). ii)-iv) The function $P_\beta(\tilde{u})$ for $\beta = 1, 0.5$ and 0.1 respectively.

fitting function of the observed data. The functional associated with this idea is given as follows:

$$\eta\mathcal{R}(\mathbf{a}) + \frac{1}{2} \int_{\Omega} \left(S(\phi) - P_\beta(\tilde{u})F(z) \right)^2 dx. \quad (8.10)$$

Our choice of regularisation for the affine parameters \mathbf{a} is:

$$\mathcal{R}(\mathbf{a}) = \frac{1}{2} \left((1 - a_1)^2 + a_2^2 + a_3^2 + a_4^2 + (1 - a_5)^2 + a_6^2 \right). \quad (8.11)$$

Other choices of \mathcal{R} are possible, and good results can also be attained with no regularisation for some examples. In particular, alternative regularisation on a_3 and a_6 based on user input is reasonable. Our method to some extent relies on a sensible placement of the initial template, especially in the case of multiple objects of a similar shape. The similarity measure we use is SSD, which we define as follows:

$$\mathcal{D}(\mathbf{a}) = \frac{1}{2} \int_{\Omega} \left(S(\phi) - P_\beta(\tilde{u})F(z) \right)^2 dx, \quad (8.12)$$

The minimisation of the affine registration step is then given as

$$\min_{\mathbf{a} \in \mathbb{R}^6} \left\{ \mathcal{J}_\eta(\mathbf{a}) = \eta\mathcal{R}(\mathbf{a}) + \mathcal{D}(\mathbf{a}) \right\}. \quad (8.13)$$

We solve this with the discretise-optimize approach, which we will briefly discuss in general terms. Using the Gauss Newton method we can update \mathbf{a} with a perturbation $\delta\mathbf{a}^{(k)}$,

$$\mathbf{a}^{(k+1)} = \mathbf{a}^{(k)} + \delta\mathbf{a}^{(k)}, \quad (8.14)$$

The Gauss Newton perturbation $\delta\mathbf{a}^{(k)}$ is given by

$$\tilde{\mathbf{H}}_{\mathcal{J}_\eta}(\mathbf{a}^{(k)})\delta\mathbf{a}^{(k)} = -\mathbf{g}_{\mathcal{J}_\eta}(\mathbf{a}^{(k)}) \quad (8.15)$$

where

$$\tilde{\mathbf{H}}_{\mathcal{J}_\eta}(\mathbf{a}^{(k)}) = \mathbf{J}^\top(\mathbf{a}^{(k)})\mathbf{J}(\mathbf{a}^{(k)}) + \eta\mathbf{H}_{\mathcal{R}}(\mathbf{a}^{(k)}) \quad (8.16)$$

and

$$\mathbf{g}_{\mathcal{J}_\eta}(\mathbf{a}^{(k)}) = \nabla_{\mathbf{a}}\mathcal{D}(\mathbf{a}) + \eta\nabla_{\mathbf{a}}\mathcal{R}(\mathbf{a}^{(k)}) \quad (8.17)$$

are the approximated Hessian and the gradient of \mathcal{J}_η at $\mathbf{a}^{(k)}$ and $\mathbf{H}_{\mathcal{R}}(\mathbf{a}^{(k)})$ are the gradient and Hessian of \mathcal{R} at $\mathbf{a}^{(k)}$ respectively. An approximated Hessian is used to deal with the nonlinearity of the problem. In this way computing higher order derivatives is avoided. Further details can be found in [41]. On convergence of this scheme we obtain $\mathbf{a}^* = \{a_1^*, a_2^*, a_3^*, a_4^*, a_5^*, a_6^*\}$. Then we define

$$\phi^*(x) = \begin{bmatrix} a_1^* & a_2^* \\ a_4^* & a_5^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} a_3^* \\ a_6^* \end{bmatrix}.$$

This allows us to define a new interior fitting term, given by $S(\phi^*)$.

8.4.2 Stage 2: Segmentation

The second stage involves using the result of the affine registration step to provide an accurate segmentation based on the relation between the shape prior and the observed data. Given the interior fitting term (i.e. $h < 0$) obtained from Stage 1, $S(\phi^*)$, we can construct an exterior fitting term (i.e. $h > 0$) from the observed data as discussed in the previous section. The complete fitting term is defined as follows:

$$h(x) = \alpha H(f)f - S(\phi^*), \quad (8.18)$$

where $\alpha > 0$ is a parameter that controls to what extent the shape prior should be balanced against the observed data. We will address the model's dependence on this choice in Section 8.5.2. This gives us a segmentation problem in the conventional convex relaxation framework, as opposed to the joint formulation (8.7) discussed earlier. This has the advantage of being a well understood problem due to the breadth of work that has addressed problems of this type [30, 18, 137, 25, 75]. Many algorithms have been proposed to solve this problem, as discussed in Chapter 3, such that finding the global minimum can be found quickly and reliably, independently of initialisation. Examples include the dual formulation [23, 18], Split Bregman [58], Chambolle-Pock [26], and Additive Operator Splitting (AOS) [85, 112, 129]. Here, we use an adjusted AOS scheme from [112], which we recall in brief from Chapter 4 next.

In the convex relaxation framework the segmentation function is constrained such that $u \in BV(\Omega; [0, 1])$. To enforce this constraint we introduce a new functional with the introduction of a regularised penalty term as discussed in [112] and earlier in the thesis, and the minimisation problem is as follows:

$$\min_u \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} h(x)u(x) dx + \mu \int_{\Omega} \nu_{\epsilon}(u) \right\}, \quad (8.19)$$

where

$$\nu_{\epsilon}(u) = H_{\epsilon} \left(\sqrt{(2u-1)^2 + \epsilon} - 1 \right) \left[\sqrt{(2u-1)^2 + \epsilon} - 1 \right],$$

and $H_{\epsilon}(x) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \frac{x}{\epsilon} \right)$. Deriving the associated Euler-Lagrange equation and introducing a time variable gives us the following PDE, in split operator form:

$$\frac{\partial u}{\partial t} = \partial_x (W \partial_x u) + \partial_y (W \partial_y u) - \lambda h - \mu \nu'_{\epsilon}(u), \quad (8.20)$$

where $W = (|\nabla u|_{\rho})^{-1}$ and is lagged to linearise the equation. Here, $|\nabla u|_{\rho} = (u_x^2 + u_y^2 + \rho)^{1/2}$ and is introduced to avoid singularities. A challenge of using AOS in the convex relaxation framework is the nonlinearity of the $\nu'_{\epsilon}(u)$ term. In order to overcome this, recalling from Chapter 4, we define an interval I_{ς} , where we adjust the equation based on the linear part of $\nu'_{\epsilon}(u)$ and the difference in u between iterations. This allows us to approximate $\nu'_{\epsilon}(u)$ in an interval, I_{ς} , with a linear function, bu . We define this interval and a binary function, \tilde{b}^n , as follows:

$$I_{\varsigma} := [0 - \varsigma, 0 + \varsigma] \cup [1 - \varsigma, 1 + \varsigma], \quad \tilde{b}^n = \begin{cases} b, & \text{if } u^n \in I_{\varsigma} \\ 0, & \text{elsewhere.} \end{cases}$$

Additionally, in Weickert et al. [129] conditions on the scheme for a discrete scale space were provided, required for convergence. The following scheme is designed to fulfil such conditions, and further details can be found in [112]. In the following the equation is rewritten in matrix-vector form after discretisation, A_{ℓ} is the diffusion quantity in the ℓ direction ($\ell = 1, 2$ for x and y directions respectively) and was derived using the finite difference method (see Chapter 5 for further details), τ is the time step size, n denotes the n^{th} iteration, and $\tilde{B}^n = \text{diag}(\tau \mu \tilde{b}^n)$. The update in each direction is given by:

$$u_{\ell}^{n+1} = \left(I - 2\tau(I + \tilde{B}^n)^{-1} A_{\ell}(u^n) \right)^{-1} (u^n - \tau(I + \tilde{B}^n)^{-1} (\mu \nu'_{\epsilon}(u^n) + \lambda f)). \quad (8.21)$$

By increasing ς , such that $\tilde{b} = b$, this scheme fulfils additional criteria from [129]. As $u \in [0, 1]$, setting $\varsigma = 0.5$ is enough to ensure this. This adjustment consists of multiplying τ by a scalar, dependent on b and μ . This can be interpreted as automatically restricting the time step, based on the prominence of the penalty function, dictated by the size of μ , and represented by b . Finally, the update at each iteration for u is given by

$$u^{n+1} = \frac{u_1^{n+1} + u_2^{n+1}}{2}. \quad (8.22)$$

Next, we briefly discuss the two-stage algorithm in full by reviewing aspects from Sections 8.4.1 and 8.4.2.

8.4.3 Two-Stage Algorithm

In this section we provide the algorithm for our two-stage fitting shape prior model, which we refer to as the FSP model from here. The observed image, which we want to segment, is denoted by z . The basis for the shape prior is a similar image (in the fitting sense) denoted by z_p , and the binary prior, u_p . The user input consists of $\tilde{\mathbf{a}}$ which ideally is the centroid of the target object in z . There are also three parameter choices: λ , α and β . Further details can be found in Sections 8.4.1 and 8.4.2. The algorithm is as follows:

Algorithm 7 Two-Stage Fitting Shape Prior Segmentation: $\Gamma^* \leftarrow$
 FSP ($z, z_p, u_p, \tilde{\mathbf{a}}, \lambda, \alpha, \beta$)

- 1: Translate binary prior based on $\tilde{\mathbf{a}}$: $\tilde{u} = u_p(\tilde{v})$
 - 2: Form shape prior: $S(x) = -H(-f(z_p))f(z_p)$
 - 3: Set Proximity function: $P_\beta(\tilde{u}) = 1 - \frac{1}{\beta} \min\{\beta, d(\tilde{u})\}$
 - 4: Determine reference for Stage 1: $F(z) = -H(-f(z))f(z)$
 - 5: **Stage 1:**
 - 6: **while** $\|\mathbf{a}^{(k)} - \mathbf{a}^{(k-1)}\| > \delta_1$ **do**
 - 7: Determine Gauss Newton perturbation $\delta\mathbf{a}^{(k)}$ using (8.15)
 - 8: Update \mathbf{a}^k using (8.14), $k = k + 1$.
 - 9: **end while**
 - 10: Determine interior fitting: $S(\phi^*)$.
 - 11: **Stage 2:**
 - 12: Set fitting function, $h(x) = \alpha H(-f(z))f(z) - S(\phi^*)$.
 - 13: Initialise $u^{(0)} = H(-h)$. Set $\mu > \|\lambda h(x)\|_{L^\infty}$ as in [30].
 - 14: **while** $\|u^{(k)} - u^{(k-1)}\| > \delta_2$ **do**
 - 15: Calculate u_1^k and u_2^k using AOS (8.21)
 - 16: Update u^k using (8.22), $k = k + 1$.
 - 17: **end while**
 - 18: $\Sigma^*(\gamma) = u^{(\ell)} > \gamma$, $\Gamma^* = \partial\Sigma^*$.
-

8.5 Experimental Results

In this section, we present three example problems and discuss results that demonstrate the advantages of our method in a number of ways. Our three test sets concern different aspects of our two-stage shape prior algorithm.

The first, Test Set 1, deals with occlusions as shown in Fig. 8.5. That is, we want to segment an object with missing or incomplete data. We present results for two different examples. The first is when boundary information of the object has been lost, and the second where the interior data is occluded. Figs. 8.6 and 8.7 demonstrate stages 1 and 2 of our algorithm respectively for the first problem, and Fig. 8.8 presents the results in full for the second problem.

The second, Test Set 2, concerns the parameter dependence of our algorithm. We

present results for a difficult medical image, shown in Fig. 8.9. We introduce an alternative formulation which has a functional analogous to the methods of [36, 130], in that its shape prior is a distance function, and its intensity fitting is a separate term. Results for our two-stage algorithm and this alternative method are given in Figs. 8.10 and 8.11 respectively. We also present a comparison of the accuracy of each method whilst varying parameters in Figs. 8.12 and 8.13.

The third, Test Set 3, is in relation to the extension of this idea to 3D. We apply our two-stage algorithm in a sequential manner, using the result on each slice as the prior for the next. We discuss the details of this approach and present some results in the context of medical imaging. The problem is defined precisely in Fig. 8.14, and results are presented for four sample slices in Figs. 8.15 and 8.16. Developing this idea, and establishing an efficient and effective way to treat 3D data involving a single 2D prior will be a direction of future work.

8.5.1 Test Set 1 (Occlusions)

In the first set of test problems, we have an image with artificial occlusions. The aim is to segment the object including the missing data, based on the shape of the prior. Here the choice of affine registration is important to preserve the desired shape. In Fig. 8.5, we have two images with different types of occlusions, and a given shape prior, $S(x)$, based on prior information z_p and u_p . We can also see the difference between the binary prior (\tilde{u}) and the boundary of the ground truth of z . These examples are similar to problems presented in [36, 130]. As these images involve single objects that can be reduced to foreground and background, we can set $\beta = 1$ for the proximity function $P_\beta(\tilde{u})$.

In Fig. 8.6, we see Stage 1 of our algorithm for Occlusion 1. We show the template and reference defined in Section 8.4.1 for the affine registration step, and the registered prior, $S(\phi^*)$. In Fig. 8.7, we see Stage 2 of our algorithm for Occlusion 2. Here, it is clear how the shape prior, $S(x)$, is related to the proposed fitting term $h(x)$. We demonstrate results of the convex relaxation stage, showing the computed contour Γ^* and the function u^* , which is approximately binary such that thresholding for $\gamma \in (0, 1)$ is legitimate. This is consistent with the ideas presented in [30, 18, 112], and discussed in Chapter 4. In Fig. 8.8 we present results for the second occlusion test problem, detailing both stages of our proposed algorithm. We can see the final result is similar to that of Occlusion 1, despite a different type of occlusion being present. We have tested other examples, varying the type and size of the occlusion, and found the method to be robust to these variations. We also note that the results are not too sensitive to the choice of parameters α and λ , which we will demonstrate in Test Set 2.

8.5.2 Test Set 2 (Parameter Dependence)

In our second set of problems we attempt to demonstrate to what extent our method is dependent on the parameters α and λ . The example we use is a medical image shown in Fig. 8.9, where we have similar images; the prior image, z_p , and the observed image,

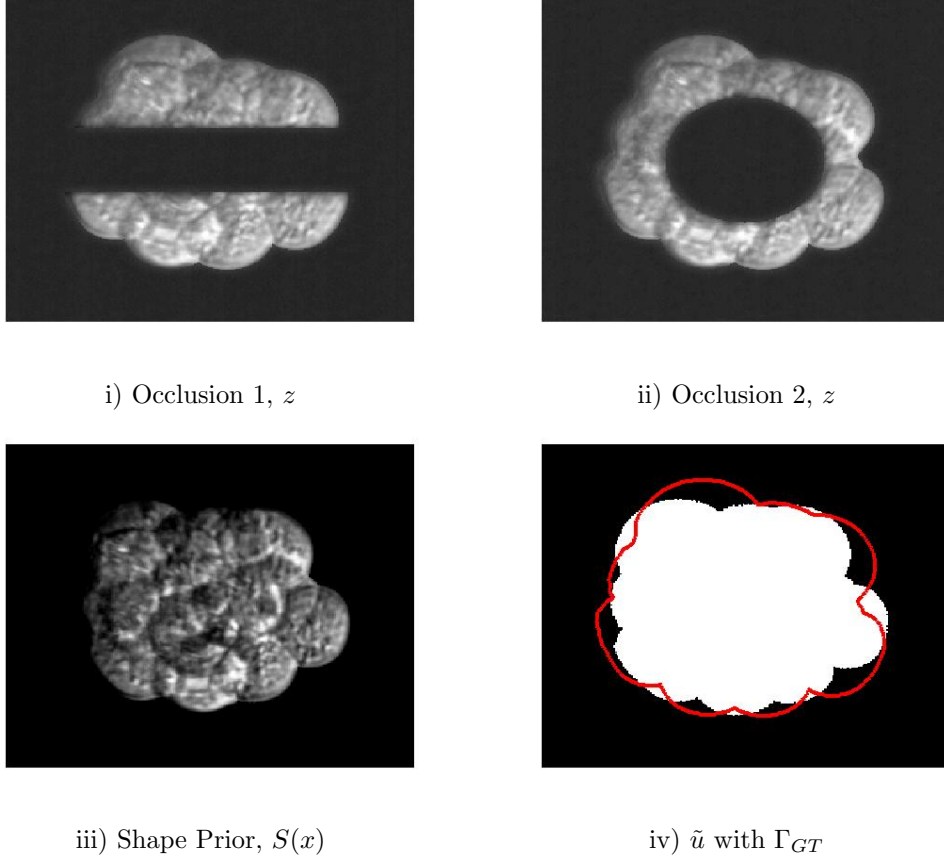


Figure 8.5: Test Set 1 (Occlusions). i) An image with an artificial occlusion. We refer to this example as Occlusion 1. ii) The same image with a different artificial occlusion. We refer to this example as Occlusion 2. iii) Shape prior term, $S(x) = -H(-f(z_p)f(z_p))$, based on our method of using the fitting term of a similar image to construct an approximate shape representation. iv) A comparison between \tilde{u} , a translation of u_p , and the boundary of the ground truth of z given by Γ_{GT} (red).

z . We can see the shape prior, $S(x)$, and how close the binary prior, \tilde{u} , is to the ground truth of z . In Fig. 8.10, we present results for our two-stage algorithm (FSP) with a shape prior for the brain. We show the result of Stage 1, the registered prior, $S(\phi^*)$, and the fitting function $h(x)$ used in Stage 2. We present the final result of FSP in the form of $u^* \in [0, 1]$ and Γ^* , which demonstrate a visually good result. However, in order to demonstrate our method's robustness to varying parameters we define the Tanimoto Coefficient [48], as in previous chapters:

$$TC = \frac{N(GT \cap \Sigma(\gamma))}{N(GT \cup \Sigma(\gamma))},$$

where GT is the ground truth, and $\Sigma(\gamma) = \{x : u(x) > \gamma\}$. By quantifying the quality of the results in this way we can clearly demonstrate how dependent our model is on parameter choice. In order to demonstrate the advantages of our method we propose comparing it to a method similar to that used by [36, 130], where a shape prior in the form of a distance function, $\psi(\tilde{u})$, is used:

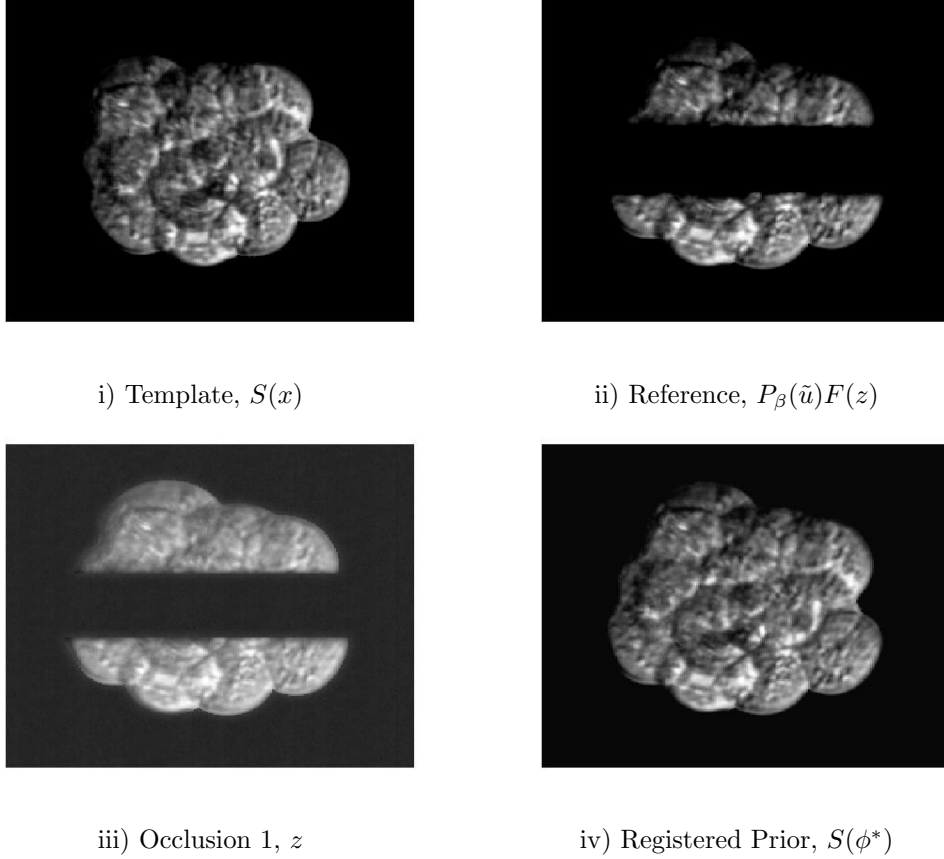
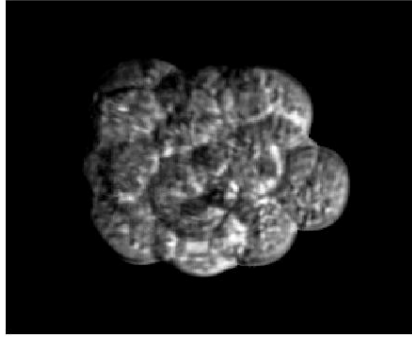


Figure 8.6: Test Set 1 (Occlusions). Stage 1: affine registration from Section 8.4.1. i) In the affine registration framework the shape prior, $S(x)$, forms the template. ii) The fitting term from the observed data, $P_\beta(\tilde{u})F(z)$, forms the reference. iii) The image, z , for Occlusion 1. iv) The result of Stage 1, where the parameters \mathbf{a}^* have been found, giving $S(\phi^*)$. This forms the basis for Stage 2.

$$\min_u \left\{ \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} f(x)u(x) dx + \theta \int_{\Omega} \psi(\tilde{u})u(x) dx \right\}. \quad (8.23)$$

This minimisation can be achieved in the same way as discussed in Section 8.4.2, and we measure this method's accuracy based on the TC when varying λ and θ . From here we will refer to (8.23) as the Distance Shape Prior (DSP) formulation. In Fig. 8.11 we present results for DSP where we show the translated binary prior, \tilde{u} , which forms the basis of the alternative shape prior, $\psi(\tilde{u})$. The final result shows $u^* \in [0, 1]$ and Γ^* , which demonstrate a visually adequate result.

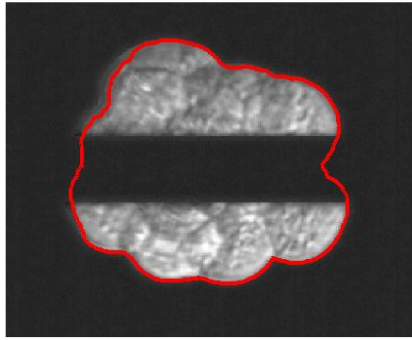
We now address the comparison of DSP with FSP, for different parameter selections. First, we show results for DSP, varying θ and λ , in Fig. 8.12. We show the shape prior, $\psi(\tilde{u})$, and three plots of $TC(\lambda)$ for $\theta = 100, 200$ and 300 . In each plot the dashed line represents the TC of \tilde{u} (which we will refer to as the Initial TC), i.e. for the algorithm to be effective it must improve on the prior information and the TC must be higher. For $\theta = 100$, we can see that in a narrow range of λ there is a minor gain in terms of accuracy. As θ increases the width of the range where an improvement is possible widens, however, the amount of gain decreases slightly for $\lambda \in [0, 300]$. This makes



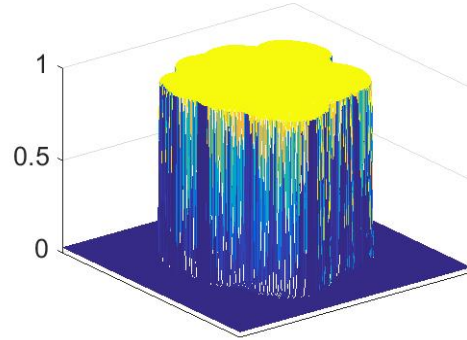
i) Shape Prior, $S(x)$



ii) Fitting, $h(x)$



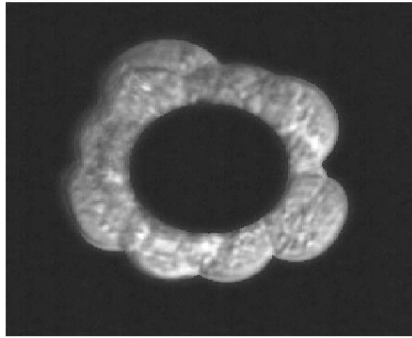
iii) Contour, Γ^*



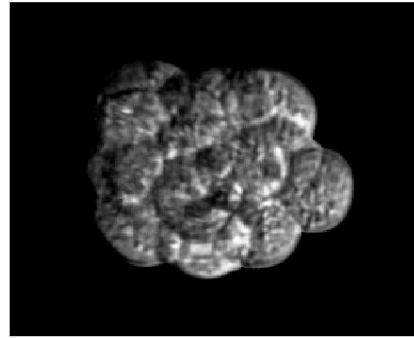
iv) $u(x) \in [0, 1]$

Figure 8.7: Test Set 1 (Occlusions). Stage 2: segmentation from Section 8.4.2. i) Shape prior term, $S(x)$, based on our method of using the fitting term of a similar image to construct an approximate shape representation. ii) The fitting term constructed from the registered prior, $S(\phi^*)$, given by $h(x)$ (8.18). iii) The computed contour, Γ^* , from Stage 2 of our algorithm. iv) The computed function $u^*(x)$ from the minimisation in the convex relaxation framework (8.19).

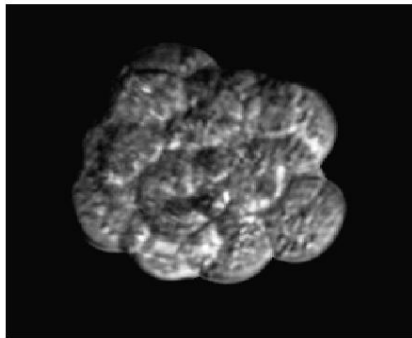
sense as if the shape term is favoured you would expect the peak TC to be closer to the Initial TC . In Fig. 8.13 we compare these results against our method, FSP. We show the shape prior, $S(x)$, where the shape approximation is given in red. For the three plots, we include the result for DSP (with $\theta = 300$) and the Initial TC , given by the dotted and dashed lines respectively. We vary α in (8.18), which controls to what extent the shape prior is favoured. A smaller α corresponds to favouring the result from Stage 1 of the algorithm. In Fig. 8.13, for $\alpha = 1$ we see a peak in TC that means a slight gain over DSP, but the accuracy does not drop off as significantly as λ increases. For $\alpha = 0.5$, we see a similar effect, with a slight improvement. However, for $\alpha = 0.1$ we see a very good result. The TC peaks significantly above that of DSP, and only drops slightly as λ increases. The TC for $\alpha = 0.1$ falls below the Initial TC for a much larger λ . this demonstrates that the selection of λ is not very sensitive, and an accurate result can be achieved for a wide range of parameters. Furthermore, we assert that the selection of α in FSP is more intuitive than θ in DSP. Our reasoning is that for FSP it is based on how visually similar the shape prior is to the observed



i) Occlusion 2, z



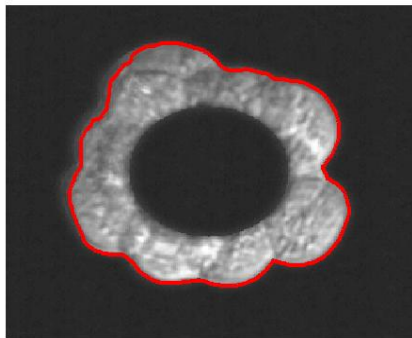
ii) Shape Prior, $S(x)$



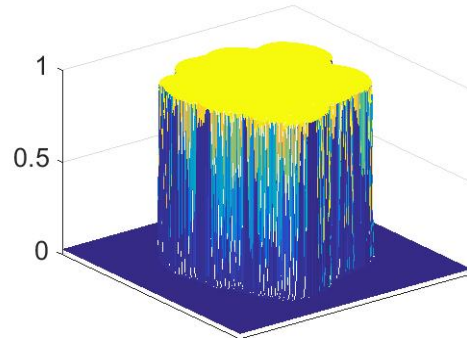
iii) Registered Prior, $S(\phi^*)$



iv) Fitting, $h(x)$



v) Contour, Γ^*



vi) $u^*(x) \in [0, 1]$

Figure 8.8: Test Set 1 (Occlusions). Stages 1 and 2 for Occlusion 2. i) The image, z , for Occlusion 2. ii) Shape prior term, $S(x)$, based on our method of using the fitting term of a similar image to construct an approximate shape representation. iii) Stage 1, where the parameters \mathbf{a}^* have been found, giving $S(\phi^*)$. iv) The fitting term constructed from $S(\phi^*)$, given by $h(x)$ (8.18). v) The computed contour, Γ^* , from Stage 2. vi) The computed function $u^*(x)$ (8.19).

data. If they are close, α should be small, and the choice of λ is not very sensitive. For DSP, the choice of θ is based on the distance of the object from others in the image, and then a difficult choice of λ .

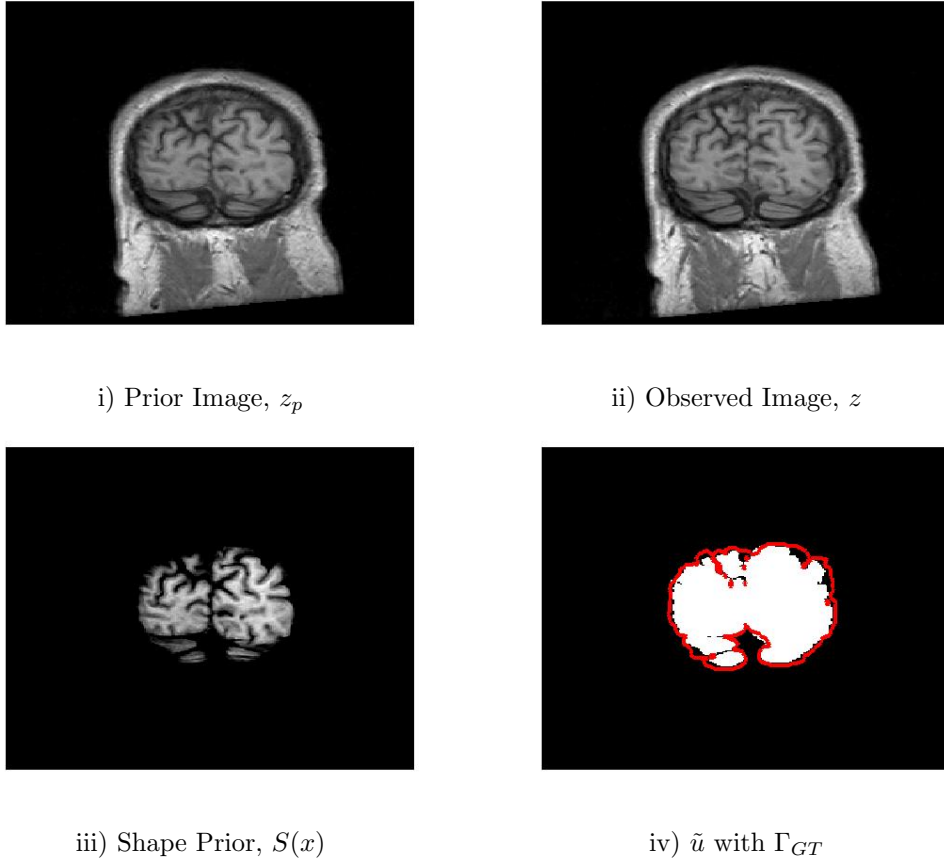
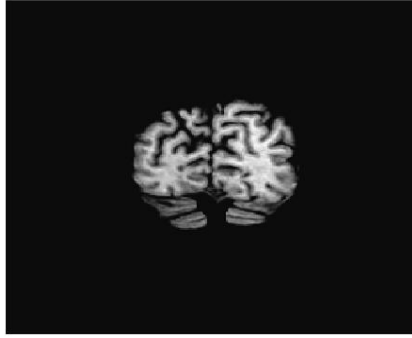


Figure 8.9: Test Set 2 (Parameter Dependence). i) The prior image, z_p , from which we know u_p . ii) The target image, z , which we want to segment based on the shape of u_p . iii) Shape prior term, $S(x) = -H(-f(z_p))f(z_p)$, based on our method of using the fitting term of a similar image to construct an approximate shape representation. iv) A comparison between \tilde{u} , a translation of u_p , and the boundary of the ground truth of z given by Γ_{GT} (red).

8.5.3 Test Set 3 (Sequential Selection)

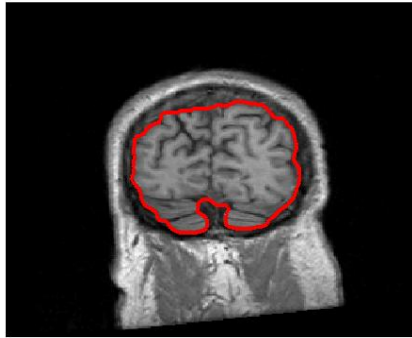
In this set of test problems we consider incorporating shape priors for a medical 3D data set, i.e. given a shape prior as defined previously on one slice can we segment the corresponding object on different slices? It is natural to consider applying our two-stage algorithm in a sequential manner in this setting. In Fig. 8.14 we present the problem we consider here. The difference between the prior image and the target image is significant, such that without using intermediate slices a result is not viable. The details of the sequential algorithm we use is given by Algorithm 8, and is simply an intuitive extension of the previous algorithm, where the priors and images used are redefined at each stage. One notes that we only provide one set of parameters (α, β, λ) and they are not refined at each slice. This is a desirable property, as the process is



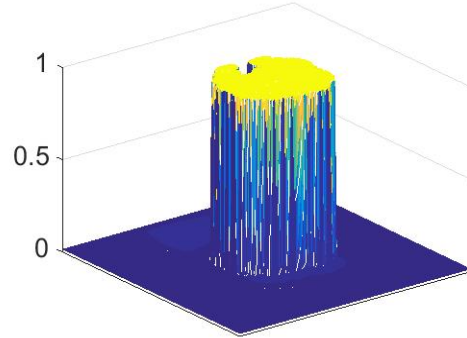
i) Stage 1, $S(\phi^*)$



ii) Stage 2, $h(x)$



iii) Contour, Γ^*

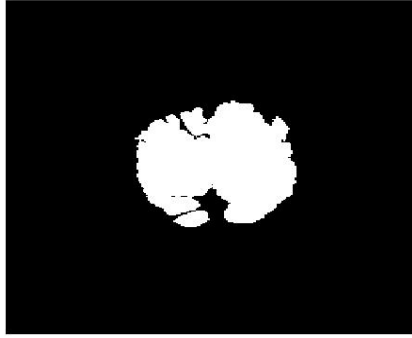


iv) $u^*(x) \in [0, 1]$

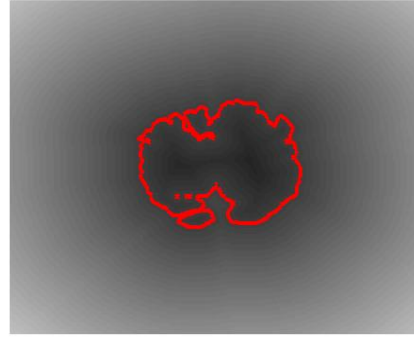
Figure 8.10: Test Set 2 (Parameter Dependence). i) The result of Stage 1 of our algorithm, where $S(\phi^*)$ is determined based on the minimisation of the affine registration formulation (8.13). ii) In Stage 2 we construct a fitting term based on the shape and intensity of the object, given by $h(x)$ (8.18). iii) The computed contour, Γ^* , from Stage 2 of our algorithm. iv) The computed function $u^*(x)$ from the minimisation in the convex relaxation framework (8.19).

therefore fully automated.

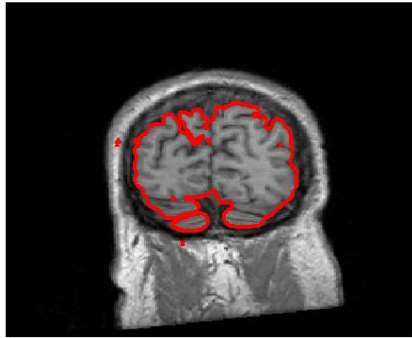
The results in Figs. 8.15 and 8.16 demonstrate impressive results in the sense that the problem presented in Fig. 8.14 was challenging. Without treating the slices in a sequential manner, this result would be beyond most conventional techniques. One notes that not every intermediate slice was used between the prior (Slice 103) and the target (Slice 123), as this would have been prohibitive from a computational time perspective. One challenge is how the slice sequence is selected. When the target object changes topology significantly it might be necessary to use more slices, and defining this precisely is challenging. Considering alternatives to this sequential method is a priority for future work. With partial information on one slice of a 3D data set, our shape representation framework (given in Section 8.2.3) should allow us to incorporate partial prior knowledge in a robust way.



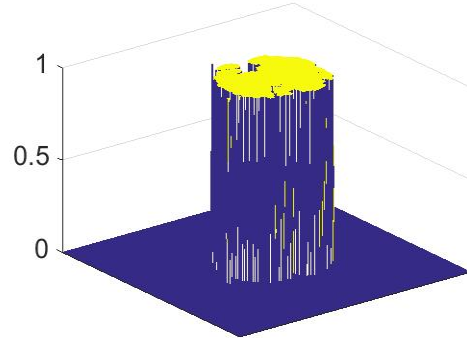
i) Binary Shape, \tilde{u}



ii) DSP Prior, $\psi(\tilde{u})$ with Γ_ψ



iii) Contour, Γ^*

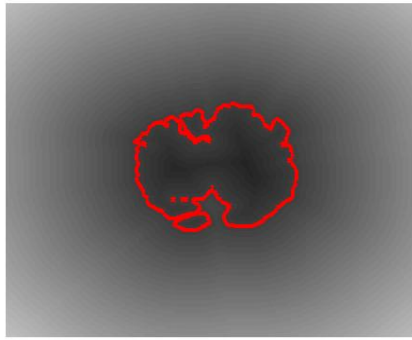


iv) $u^*(x) \in [0, 1]$

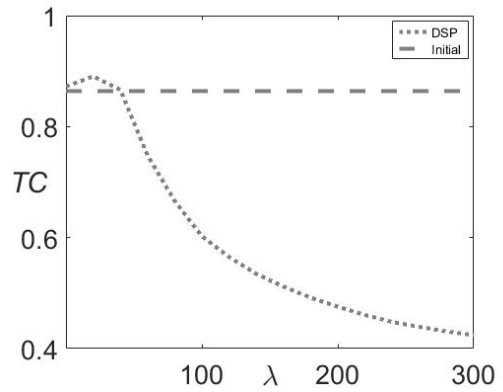
Figure 8.11: Test Set 2 (Parameter Dependence). Results obtained using DSP formulation. i) The binary shape, \tilde{u} , which is the ground truth of z_p . ii) An alternative prior, $\psi(\tilde{u})$, based on the Euclidean distance from the boundary of the translated prior. This term is similar to shape representations in [36, 130]. Here, $\Gamma_\psi = \{x : \psi(x) = 0\}$ and is shown in red. The computed contour, Γ^* , using DSP. iv) The computed function $u^*(x)$ from the minimisation problem of DSP (8.23).

8.6 Remarks

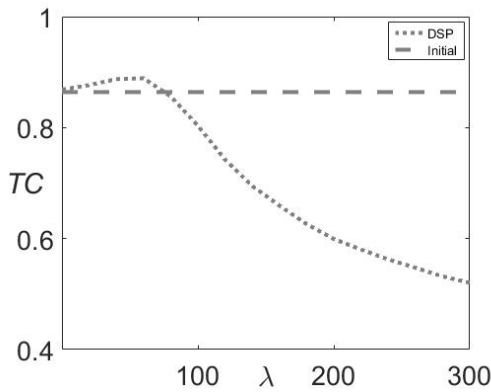
In this chapter we have provided a brief review of shape prior segmentation methods, and discussed the motivation for a new two-stage algorithm for incorporating shape priors in variational segmentation. The central idea is based on approximating shapes with intensity fitting functions and using affine registration to compare similar images. An improved fitting term is constructed based on the shape prior, where the intensity information is implicitly enforced. This approach is demonstrated to be robust to parameter changes compared to conventional methods, and is capable of achieving good results for a range of examples, including significant occlusions. Finally, we present an object selection example for medical images and discuss its potential application to 3D segmentation.



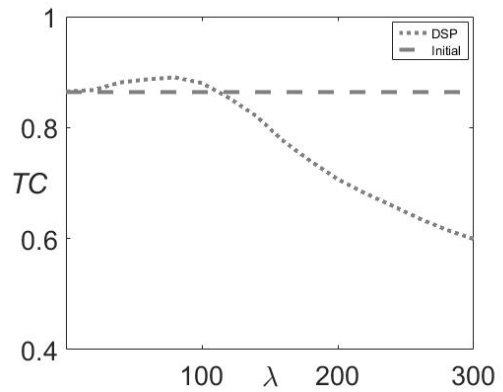
i) DSP Prior, $\psi(\tilde{u})$ with Γ_ψ



ii) $\theta = 100$



iii) $\theta = 200$



iv) $\theta = 300$

Figure 8.12: Test Set 2 (Parameter Dependence). DSP compared against initial TC of \tilde{u} . i) Shape prior, $\psi(x)$, used in (8.23). Here, $\Gamma_\psi = \{x : \psi(x) = 0\}$ and is shown in red. ii) - iv) $TC(\lambda)$ for different choices of θ in DSP, and the initial TC of \tilde{u} . Varying $\lambda \in [0, 300]$ gives some improvement over the initial TC . As θ increases, the range of λ that offers an improvement gets larger. However, the extent of this improvement is also lessened as θ increases for $\lambda \in [0, 300]$. This makes sense as the $\psi(x)$ term favours \tilde{u} . Balancing λ and θ with DSP can be challenging, and offers limited improvements over the given prior.

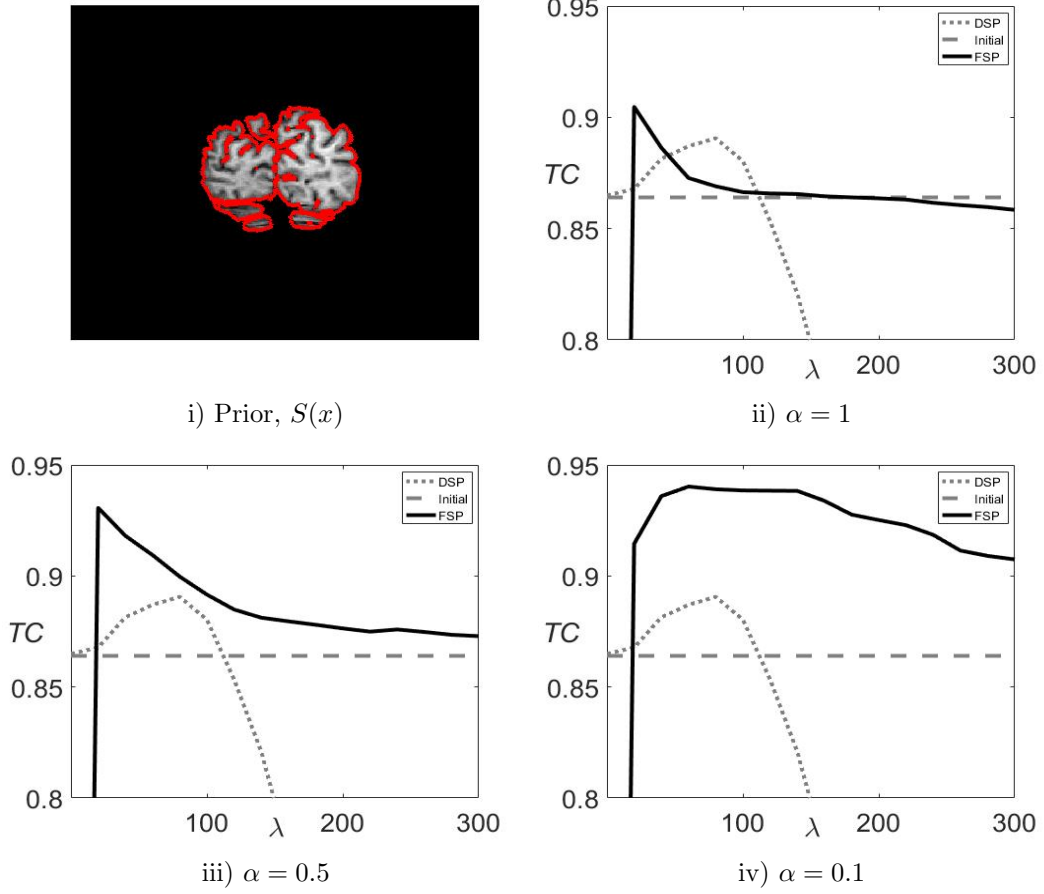


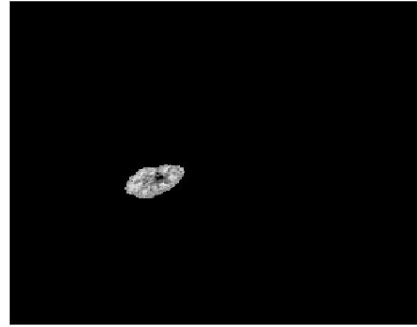
Figure 8.13: Test Set 2 (Parameter Dependence). Two-Stage Fitting Shape Prior Model (FSP) compared against alternative DSP and the initial TC of \tilde{u} . i) Shape prior, $S(x)$, used in (8.10). ii) - iv) $TC(\lambda)$ for different choices of α in FSP against DSP (with $\theta = 300$), and TC of \tilde{u} . For $\alpha < 1$, TC is consistently above the initial TC , and peaks higher than DSP does for any λ, θ pair. For $\alpha = 0.1$, we can see that we have a substantial gain over DSP, both in terms of the optimal choice and the dependence on the parameters selection.

Algorithm 8 Sequential FSP: $\Gamma^* \leftarrow \text{SFSP}(z^{(1, \dots, n)}, z_p, u_p, \tilde{\mathbf{a}}, \lambda, \alpha, \beta)$

- 1: Translate binary prior based on $\tilde{\mathbf{a}}$: $\tilde{u} = u_p(\tilde{v})$
 - 2: **for** $\ell = 1 : n$
 - 3: Form shape prior: $S(x) = -H(-f(z_p))f(z_p)$
 - 4: Set Proximity function: $P_\beta(\tilde{u}) = 1 - \frac{1}{\beta} \min\{\beta, d(\tilde{u})\}$
 - 5: Determine reference for Stage 1: $F(z) = -H(-f(z))f(z)$
 - 6: **Stage 1:**
 - 7: Determine interior fitting: $S(\phi^*)$.
 - 8: **Stage 2:**
 - 9: Set fitting function, $h(x) = \alpha H(-f(z))f(z) - S(\phi^*)$.
 - 10: Initialise $u^{(0)} = H(-h)$. Set $\mu > \|\lambda h(x)\|_{L^\infty}$ as in [30].
 - 11: **while** $\|u^{(k)} - u^{(k-1)}\| > \delta_2$ **do**
 - 12: Update u^k using (8.22), $k = k + 1$.
 - 13: **end while**
 - 14: $z_p = z$, $\tilde{u} = u^k$, $z = z^{(\ell+1)}$
 - 15: **end for**
 - 16: $\Sigma^*(\gamma) = u^{(k)} > \gamma$, $\Gamma^* = \partial \Sigma^*$.
-



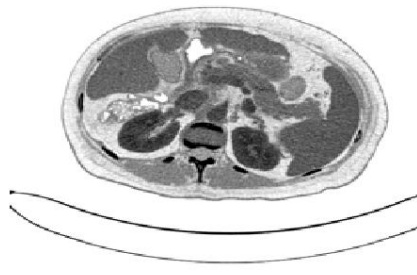
i) Prior Image, $z_p(x)$



ii) Shape Prior, $S(x)$

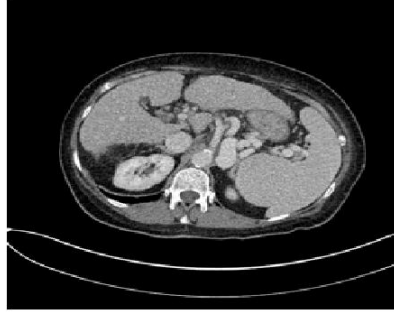


iii) Target Image, $z(x)$

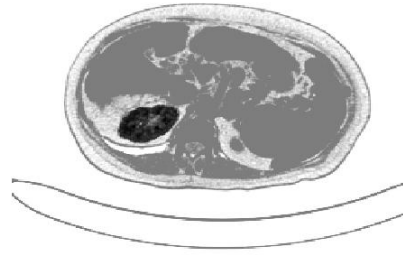


iv) Target Fitting, $f(x)$

Figure 8.14: Test Set 3 (Sequential Selection). Problem Definition: Given a prior image, $z_p(x)$, and a corresponding shape prior, $S(x)$, according to Section 8.4 given by i) and ii) respectively, we aim to successfully segment the same object in a different slice of a 3D data set. iii) gives the target image, $z(x)$, and iv) gives the fitting term of $z(x)$. We can achieve a result by applying our proposed two-stage model to the intermediate slices, which is defined in detail in Algorithm 8.



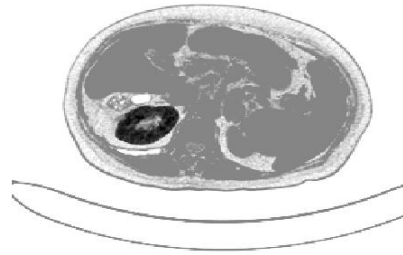
i) Slice 107, $z(x)$



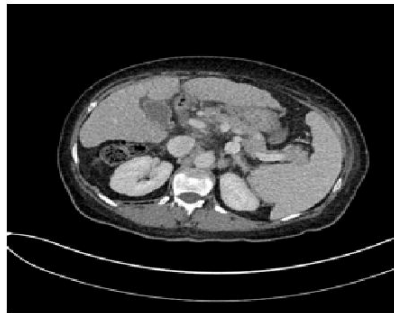
ii) Slice 107, $h(x)$



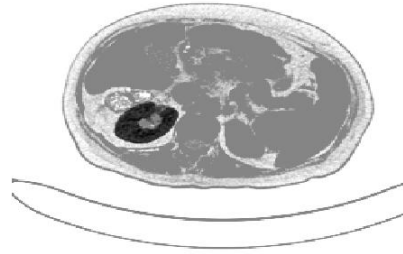
iii) Slice 112, $z(x)$



iv) Slice 112, $h(x)$



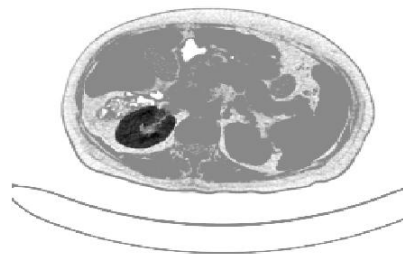
v) Slice 118, $z(x)$



vi) Slice 118, $h(x)$

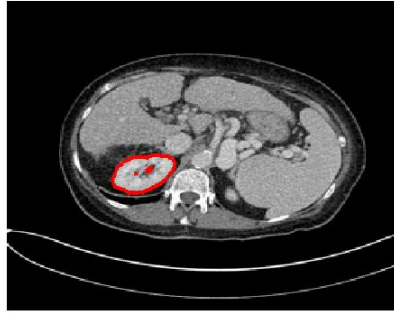


vii) Slice 123, $z(x)$



viii) Slice 123, $h(x)$

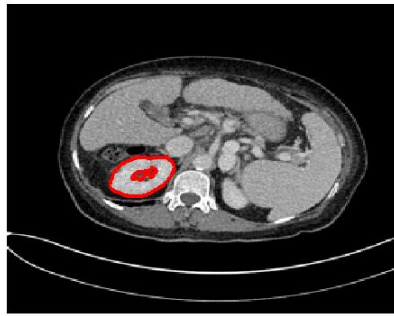
Figure 8.15: Test Set 3 (Sequential Selection). Stage 1 Results. i) The image $z(x)$ at Slice 107 of the set. ii) The fitting function $h(x)$ determined from Stage 1 of the algorithm for Slice 107. Similar for iii)-iv) Slice 112, v)-vi) Slice 118, and vii)-viii) Slice 123.



i) Slice 107, $z(x)$, Γ^*



ii) Slice 107, $u^*(x)$



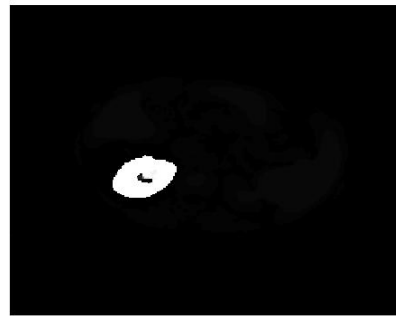
iii) Slice 112, $z(x)$, Γ^*



iv) Slice 112, $u^*(x)$



v) Slice 118, $z(x)$, Γ^*



vi) Slice 118, $u^*(x)$



vii) Slice 123, $z(x)$, Γ^*



viii) Slice 123, $u^*(x)$

Figure 8.16: Test Set 3 (Sequential Selection). Stage 2 Results. i) The computed contour Γ^* for Slice 107. ii) The segmentation function, $u^*(x)$, determined from Stage 2 of the algorithm. Similar for iii)-iv) Slice 112, v)-vi) Slice 118, and vii)-viii) Slice 123.

Chapter 9

Conclusions and Future Work

The work in this thesis concerns variational methods for image segmentation. In early chapters we briefly introduce some relevant mathematical preliminaries and related variational methods for image processing. In particular we focus on important foundational works in image segmentation, such as Mumford and Shah [89] and Active Contours Without Edges [33]. We also highlight convex relaxation methods and discuss associated algorithms, as these ideas are central to our work. In Chapter 4 we introduce a new method for computing global minimisers of two-phase segmentation problems with a generalised fitting function. Our main contribution here is an additive operator splitting (AOS) scheme for globally convex segmentation, and our results compare favourably to Chambolle’s dual formulation [23, 18] and is an improvement on the original scheme [30, 17]. With this work as a foundation, in subsequent chapters we introduce work aimed to address various areas within variational methods for image segmentation. In brief, this includes work on intensity inhomogeneity, image reconstruction, and incorporating user input and priors. Our experimental results demonstrate the effectiveness of our proposed methods, particularly with respect to reliability. In Chapters 4, 6, and 8 we demonstrate a reduced dependence on the fitting parameter λ compared to similar methods. This is significant as current approaches tend to require this parameter to be selected empirically, and therefore results are often heavily reliant on user selection. In the following, we offer some concluding remarks in relation to each chapter and address areas for future work.

In Chapter 4 we tested our proposed AOS method against an analogous method [23, 18], i.e. the relaxed binary constraint is imposed with a penalty function. Our method performs well compared with this dual formulation, with encouraging results in terms of accuracy, computation time, and how close to binary the solution is. The results in relation to our improved AOS method are supported by additional results in Chapter 5. Importantly, we also demonstrate that our approach appears to be less dependent on the fitting parameter, λ . This is a significant finding in the sense that in practice it makes our method more reliable. Future work could consist of investigating how the AOS interval can be defined in a consistent way, as discussed in Chapter 4. Further theoretical work on the proposed AOS schemes could help resolve this matter. There are clear advantages in extending this work to the 3D case, for a generalised

fitting function. AOS extends to multiple dimensions easily, and potentially offers advantages over current work [23, 18]. It is also worth comparing the proposed method against alternatives, such as Split Bregman [58] or Chambolle-Pock [25].

In Chapter 5 we discussed the drawbacks of current selective segmentation models and proposed a new model where a global minimiser can be found independently of initialisation. The considerations of [30] and [105] motivated a proposed nonconvex selective model, and we detail its convex reformulation. Our nonconvex model performs well in comparison to Rada-Chen [105], and we demonstrate that global minimisers of the corresponding nonconvex formulation can be found. In our results we show that simple user input (i.e. three or four markers) can produce a good result in a difficult example and, importantly, a similar result can be obtained for a very different marker set. The results we provide here demonstrate the potential applications of selective segmentation. No knowledge of the object is known a priori and yet we can achieve satisfactory results for difficult examples from medical imaging. We extend this concept to intensity inhomogeneity in Chapter 6, with positive results in this case as well. Future work could involve employing alternative fitting functions, based on different assumptions about the image intensity. Also important to consider is the selection parameter θ . Currently, it is chosen empirically based on the difference between the user input and the target object. Reducing any sensitivity is essential for possible applications.

In Chapter 6 we introduced a constraint to the Variant Mumford-Shah Model [37], although it applies to any model using the bias field framework for two-phase segmentation. It is an improvement over existing methods in the sense that the intensity constants reliably converge and are feasible in relation to the image. We also observe possible advantages with this framework in terms of the quality of the piecewise-smooth approximation of the image, and a model less reliant on the fitting parameter. This is a potentially important finding, as this 'stabilisation' of the bias field appears to allow for more parameter variation thus improving the reliability of the models. We will investigate this idea further in the future, and attempt to accurately quantify an improvement. It is important to note that whilst the intensity constants converge to values consistent with the image, the model is not jointly convex. This means that the computed c_1 and c_2 are not necessarily the true values, and are dependent on initialisation. We could extend the work of Brown et al. [20] to the bias field framework and address possible contradictions with these formulations.

In Chapter 7 we address the problem of segmentation with a low quality observed image. We proposed a joint model for the reconstruction and segmentation of blurred images where the blur function is unknown, which we call blind image segmentation. The results presented demonstrate advantages for treating this problem simultaneously, rather than in a two-stage manner. Also, we introduce a relaxed model that reduces computation time based on alternating direction methods. In terms of future work, it is possible to consider using the improved AOS methods introduced in Chapter 4 in order to improve performance further. We compare our models to the work of Bar et al.

[11], but future work could consider quantitative comparisons with alternative methods [107, 28, 97]. This chapter addresses a fundamental problem in image segmentation, as many observed images aren't of sufficient quality to be treated in a conventional way [30, 18]. Further refinements to the formulation could yield important results, with many potential applications.

In the last chapter we incorporate shape priors in two-phase globally convex segmentation. We review previous shape based segmentation methods, and discuss the motivation for a new two-stage model which is shown to be robust to parameter variation compared to conventional approaches. Future work could consist of considering ways to implement our method in a joint formulation, which could improve results. The central idea is based on approximating shapes using intensity fitting functions, and future work could consider extending this to alternative fitting functions such as Nguyen et al. [92]. Also, alternatives to affine registration could be considered such as Cremers et al. [39] who employ a template based formulation. One area we have addressed in Chapter 8 is to do with extending these ideas to 3D segmentation, based on a single 2D shape prior. In Section 8.5.3 we propose Algorithm 8 for sequential segmentation. Future work could involve how to address 3D data more efficiently by considering the entire sequence simultaneously.

Bibliography

- [1] R. Adams. *Sobolev Spaces*. Academic Press, 1975.
- [2] M. Ahmed, S. Yamany, N. Mohamed, A. Farag, and T. Moriarty. A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. *IEEE Transactions on Medical Imaging*, 21(3):193–199, 2002.
- [3] H. Ali, N. Badshah, K. Chen, and G. Khan. A variational model with hybrid images data fitting energies for segmentation of images with intensity inhomogeneity. *Pattern Recognition*, 51:27–42, 2016.
- [4] M. Almeida and L. Almeida. Blind and semi-blind deblurring of natural images. *IEEE Transactions on Image Processing*, 19(1):36–52, 2010.
- [5] L. Ambrosio and V.M. Tortorelli. On the approximation of free discontinuity problems. *Bollettino dell’Unione Matematica Italiana*, 7(6-B):105–123, 1992.
- [6] H. Anton. *Elementary Linear Algebra*. Wiley, 1973.
- [7] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer, 2006.
- [8] N. Badshah and K. Chen. Image selective segmentation under geometrical constraints using an active contour approach. *Communications in Computational Physics*, 7(4):759–778, 2009.
- [9] N. Badshah, K. Chen, H. Ali, and G. Murtaza. Coefficient of variation based image selective segmentation model using active contours. *East Asian Journal on Applied Mathematics*, 2:150–169, 2012.
- [10] E. Bae, J. Yuan, and X. C. Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision*, 92:112–129, 2011.
- [11] L. Bar, N. Sochen, and N. Kiryati. Variational pairing of image segmentation and blind restoration. In *Proceedings of the 8th European Conference on Computer Vision*, volume 3022 of *Lecture Notes in Computer Science*, pages 166–177. Springer, 2004.

- [12] L. Bar, N. Sochen, and N. Kiryati. Semi-blind image restoration via Mumford-Shah regularization. *IEEE Transactions on Image Processing*, 15(2):483–493, 2006.
- [13] M. Barchiesi, S. H. Kang, T. Le, M. Morini, and M. Ponsiglione. A variational model for infinite perimeter segmentations based on lipschitz level set functions: denoising while keeping finely oscillatory boundaries. *Multiscale Modeling and Simulation*, 8(5):1715–1741, 2010.
- [14] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.
- [15] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222–1239, 2001.
- [16] K. Bredies, K. Kunich, and T. Pock. Total generalized version. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.
- [17] X. Bresson. *Image Segmentation with Variational Active Contours*. PhD thesis, École Supérieure d’Électricité, 2005.
- [18] X. Bresson, S. Esedoglu, P. Vanderghenst, J. P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, 2007.
- [19] E. Brown, T. Chan, and X. Bresson. A convex relaxation method for a class of vector-valued minimization problems with applications to Mumford-Shah segmentation. *UCLA CAM Report*, 10-44, 2010.
- [20] E. Brown, T. Chan, and X. Bresson. Completely convex formulation of the Chan-Vese image segmentation model. *International Journal of Computer Vision*, 98:103–121, 2012.
- [21] T. Brox and D. Cremers. On local region models and a statistical interpretation of the piecewise smooth Mumford-Shah functional. *International Journal of Computer Vision*, 84(2):184–193, 2009.
- [22] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
- [23] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, 2004.
- [24] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An introduction to total variation for image analysis. In *Theoretical Foundations and Numerical Methods for Sparse Recovery*, volume 9 of *Radon Series on Computational and Applied Mathematics*, pages 263–340. De Gruyter, 2010.

- [25] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM Journal in Imaging Sciences*, 5(4):1113–1158, 2012.
- [26] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011.
- [27] R. Chan, M. Tao, and X. Yuan. Constrained total variational deblurring models and fast algorithms based on alternating direction method of multipliers. *SIAM Journal on Imaging Sciences*, 6(1):680–697, 2013.
- [28] R. Chan, H. Yang, and T. Zeng. A two-stage image segmentation method for blurry images with poisson or multiplicative gamma noise. *SIAM Journal on Imaging Sciences*, 7(1):98–127, 2014.
- [29] T. Chan, S. Esedoglu, and M. Nikolova. Algorithm for finding global minimizers of image segmentation and denoising models. *UCLA CAM Report*, 04-54, 2004.
- [30] T. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.
- [31] T. Chan and J. Shen. *Image Processing and Analysis*. SIAM, 2005.
- [32] T. Chan and J. Shen. Variational image inpainting. *Communications on Pure and Applied Mathematics*, 58(5):579–619, 2005.
- [33] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.
- [34] T. Chan and L. Vese. A level set algorithm for minimizing the Mumford-Shah functional in image processing. In *Proceedings of the IEEE Workshop On Variational and Level Set Methods*, pages 161–168, 2001.
- [35] T. Chan and C. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–375, 1998.
- [36] T. Chan and W. Zhu. Level set based shape prior segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1170, 2005.
- [37] D. Chen, M. Yang, and L. Cohen. Global minimum for a variant Mumford-Shah model with application to medical image segmentation. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 1(1):48–60, 2013.
- [38] K. Chen, S. Harding, B. Williams, and Y. Zheng. A new study of blind deconvolution with implicit incorporation of non-negativity constraints. *International Journal of Computational Mathematics*, 2015.

- [39] S. Chen, D. Cremers, and R. Radke. Image segmentation with one shape prior - A template-based formulation. *Image and Vision Computing*, 30(12):1032–1042, 2012.
- [40] Y. Chen, H. Thiruvenkadam, H. Tagare, F. Huang, D. Wilson, R. Briggs, K. Gopinath, and E. Geiser. On the incorporation of shape priors into geometric active contours. *International Journal of Computer Vision*, 50(3):315–328, 2002.
- [41] N. Chumchob and K. Chen. A robust affine image registration method. *International Journal of Numerical Analysis and Modeling*, 6(3):311–334, 2009.
- [42] G. Chung. *A Unifying Framework for Piecewise-Constant Image Segmentation and Deblurring*. PhD thesis, University of California, Los Angeles, 2007.
- [43] G. Chung and L. Vese. Image segmentation using a multilayer level-set approach. *Computing and Visualization in Science*, 12(6):267–285, 2009.
- [44] M. Crandall, M. Ishii, and H. Lions. Users’ guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27(1):1–69, 1992.
- [45] D. Cremers, F. Schmidt, and F. Barthel. Shape priors in variational image segmentation: Convexity, lipschitz continuity and globally optimal solutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2008.
- [46] D. Cremers, C. Schnörr, and J. Weickert. Diffusion Snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.
- [47] D. Cremers, N. Sochen, and C. Schnörr. Towards recognition-based variational segmentation using shape priors and dynamic labelling. In *Scale Space and Variational Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 388–400. Springer, 2003.
- [48] W. Crum, O. Camara, and D. Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE Transactions on Medical Imaging*, 25(11):1451–1461, 2006.
- [49] Y. Duan, H. Chang, W. Huang, and J. Zhou. Simultaneous bias correction and image segmentation via L0 regularized Mumford-Shah model. In *Proceedings of the IEEE International Conference on Image Processing*, pages 6–40, 2014.
- [50] Y. Duan, Y. Wang, X. C. Tai, and J. Hahn. A fast augmented lagrangian method for Euler’s elastica model. In *Scale Space and Variational Methods in Computer Vision*, volume 6667 of *Lecture Notes in Computer Science*, pages 144–156. Springer, 2012.

- [51] I. Ekeland and R. Témam. *Convex Analysis and Variational Problems*. SIAM, 1999.
- [52] H. Federer. *Geometric Measure Theory*. Springer, 1969.
- [53] W. Fleming and R. Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11(1):218–222, 1960.
- [54] I. Gelfand and S. Fomin. *Calculus of Variations*. Prentice-Hall, Inc., 1963.
- [55] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- [56] M. Giaquinta and S. Hildebrandt. *Calculus of Variations I, The Lagrangian Formalism*. Springer-Verlag, 1996.
- [57] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*, volume 80 of *Monographs in Mathematics*. Birkhauser Boston, 1984.
- [58] T. Goldstein, X. Bresson, and S. Osher. Geometric applications of the split bregman method. *Journal of Scientific Computing*, 45(1-3):272–293, 2010.
- [59] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
- [60] C. Gout and C. Le Guyader. Geodesic active contour under geometrical conditions: Theory and 3D Applications. *Numerical Algorithms*, 48:105–133, 2008.
- [61] C. Gout, C. Le Guyader, and L. Vese. Segmentation under geometrical conditions with geodesic active contours and interpolation using level set methods. *Numerical Algorithms*, 39:155–173, 2005.
- [62] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51:271–279, 1989.
- [63] J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 1902. 49-52.
- [64] N. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002.
- [65] J. B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms. I. Fundamentals*. Grundlehren der mathematischen Wissenschaften 305. Springer-Verlag, 1993.
- [66] M. Ibrahim, K. Chen, and C. Brito-Loeza. A novel variational model for image registration using gaussian curvature. *Journal of Geometry, Imaging and Computing*, 1(4):417–446, 2014.

- [67] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.
- [68] H. Ishikawa and D. Geiger. Segmentation by grouping functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–131, 1998.
- [69] M. Jung, G. Chung, G. Sundaramoorthi, L. Vese, and A. Yuille. Sobolev gradients and joint variational image segmentation, denoising, and deblurring. In *Proceedings of SPIE 7246, Computational Imaging VII*, 2009.
- [70] M. Jung, G. Peyre, and L. Cohen. Nonlocal active contours. *SIAM Journal on Imaging Sciences*, 5(3):1022–1054, 2012.
- [71] L. M. Lui K. C. Lam. Landmark and intensity based registration with large deformations via quasi-conformal maps. *SIAM Journal on Imaging Sciences*, 7(4):2364–2392, 2014.
- [72] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [73] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Conformal curvature flows: from phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134:275–301, 1996.
- [74] J. Kim, A. Tsai, M. Cetin, and A. Willsky. A curve evolution-based variational approach to simultaneous image restoration and segmentation. In *Proceedings of the IEEE International Conference on Image Processing*, pages 109–112, 2002.
- [75] M. Klodt and D. Cremers. A convex framework for image segmentation with moment constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011.
- [76] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:147–159, 2004.
- [77] S. Lanktona and A. Tannenbaum. Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, 2008.
- [78] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex multi-class image labeling by simplex-constrained total variation. In *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 150–162. Springer, 2009.
- [79] J. Lellmann and C. Schnörr. Continuous multiclass labeling approaches and algorithms. *SIAM Journal on Imaging Sciences*, 4(4):1049–1096, 2011.

- [80] M. Leventon, E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 316–322, 2000.
- [81] C. Li, R. Huang, J. Gatenby, D. Metaxas, and J. Gore. A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI. *IEEE Transactions on Image Processing*, 20(7):2007–2016, 2011.
- [82] C. Li, C. Kao, J. Gore, and Z. Ding. Minimization of region-scalable fitting energy for image segmentation. *IEEE Transactions on Image Processing*, 17(10):1940–1949, 2008.
- [83] C. Li, C. Xu, C. Gui, and M. Fox. Level set evolution without re-initialization: A new variational formulation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 430–436, 2005.
- [84] F. Li, M. Ng, and C. Li. Variational fuzzy Mumford-Shah model for image segmentation. *SIAM Journal on Applied Mathematics*, 70(7):2750–2770, 2010.
- [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(6):673–708, 1992.
- [86] S. Makni, P. Ciuciu, J. Idier, and J. B. Poline. Semi-blind deconvolution of neural impulse response in fMRI using a Gibbs sampling method. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 601–604, 2004.
- [87] A. Mitiche and I. Ben Ayed. *Variational and Level Set Methods in Image Segmentation*, volume 5 of *Springer Topics in Signal Processing*. Springer, 2010.
- [88] J. Modersitzki. *Numerical Methods for Image Registration*. Oxford University Press, 2004.
- [89] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.
- [90] E. Mylonas, M. Savelonas, and D. Maroulis. Automated adjustment of region-based active contour parameters using local image geometry. *IEEE Transactions on Cybernetics*, 44(12):2757–2770, 2014.
- [91] M. Ng, G. Qiu, and A. Yip. Numerical methods for interactive multiple class image segmentation problems. *International Journal of Imaging Systems and Technology*, 20(3):191–201, 2010.
- [92] T. Nguyen, J. Cai, J. Zhang, and J. Zheng. Robust interactive image segmentation using convex active contours. *IEEE Transactions on Image Processing*, 21:3734–3743, 2012.

- [93] M. Nielsen, L. Florack, and R. Deriche. Regularization and scale space. Technical report, INRIA, 1994.
- [94] J. Ortega and W. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM, 1970.
- [95] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.
- [96] S. U. Park, N. Dobigeon, and A. Hero. Semi-blind sparse image reconstruction with application to mrfm. *IEEE Transactions on Image Processing*, 21(9):3838–3849, 2012.
- [97] G. Paul, J. Cardinale, and I. Sbalzarini. Coupling image restoration and segmentation: A generalized linear model/bregman perspective. *International Journal of Computer Vision*, 104(1):69–93, 2013.
- [98] N. Persch, A. Elhayek, M. Welk, A. Bruhn, S. Grewenig, K. Böse, A. Kraegeloh, and J. Weickert. Enhancing 3D cell structures in confocal and STED microscopy: a joint model for interpolation, deblurring and anisotropic smoothing. *Measurement Science and Technology*, 24(12):125703, 2013.
- [99] G. Pillonetto and C. Cobelli. Identifiability of the stochastic semi-blind deconvolution problem for a class of time-invariant linear systems. *Automatica*, 43(4):647–654, 2007.
- [100] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.
- [101] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. A convex relaxation approach for computing minimal partitions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 810–817, 2009.
- [102] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *Proceedings of the 10th European Conference on Computer Vision*, volume 5304 of *Lecture Notes in Computer Science*, pages 792–805. Springer, 2008.
- [103] R. Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.
- [104] L. Rada and K. Chen. A new variational model with dual level set functions for selective segmentation. *Communications on Computational Physics*, 12(1):261–283, 2012.
- [105] L. Rada and K. Chen. Improved selective segmentation model using one level-set. *Journal of Algorithms & Computational Technology*, 7(4):509–540, 2013.

- [106] L. Rada and K. Chen. On a variational model for selective image segmentation of features with infinite perimeter. *Journal of Mathematical Research with Applications*, 33(3):253–272, 2013.
- [107] R. Reddy, M. Chandra, and R. Rao. Segmentation of blurred images using improved chan-vese snake model. In *International Conference on Signal Processing, Communication, Computing and Networking Technologies*, pages 502–505, 2011.
- [108] M. Rousson and N. Paragios. Shape priors for level set representations. In *Proceedings of the 7th European Conference on Computer Vision*, volume 2351 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2002.
- [109] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1):259–268, 1992.
- [110] P. Sarri, G. Thomas, E. Sekko, and P. Neveux. Myopic deconvolution combining kalman filter and tracking control. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1833–1836, 1998.
- [111] J. Spencer and K. Chen. Advanced methods in variational learning: Segmentation with intensity inhomogeneity. In *Mathematical Problems in Data Science*, pages 171–187. Springer, 2015.
- [112] J. Spencer and K. Chen. A convex and selective variational model for image segmentation. *Communications in Mathematical Sciences*, 13(6):1453–1472, 2015.
- [113] J. Spencer and K. Chen. Dynamic distance selective segmentation. Challenges in Dynamic Imaging Data, Isaac Newton Institute, 2015.
- [114] J. Spencer and K. Chen. Global and local segmentation of images by geometry preserving variational models and their algorithms. In *Forging Connections between Computational Mathematics and Computational Geometry*, pages 87–105. Springer, 2015.
- [115] J. Spencer and K. Chen. Selective image segmentation with intensity inhomogeneity. 26th Biennial Numerical Analysis Conference, University of Strathclyde, 2015.
- [116] J. Spencer and K. Chen. Shape prior segmentation with intensity inhomogeneity. New Directions in Numerical Computation, University of Oxford, 2015.
- [117] J. Spencer and K. Chen. Image segmentation with a shape prior. SIAM Imaging Science, Albuquerque, 2016.
- [118] J. Spencer and K. Chen. Stabilised bias field: Segmentation with intensity inhomogeneity. *Journal of Algorithms & Computational Technology*, 2016. submitted.
- [119] G. Strang. Maximal flow through a domain. *Mathematical Programming*, 26(2):123–143, 1983.

- [120] E. Strelakovsky and D. Cremers. Generalized ordering constraints for multilabel optimization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011.
- [121] A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Halsted Press, 1977.
- [122] A. Tsai, A. Yezzi, and A. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.
- [123] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.
- [124] C. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.
- [125] C. Vogel and M. E. Oman. Fast total variation-based image reconstruction. In *Proceedings of the ASME Design Engineering Conferences*, volume 3(C), pages 1009–1015, 1995.
- [126] F. Wang. *Alternating Direction Methods for Image Recovery*. PhD thesis, Hong Kong Baptist University, 2013.
- [127] J. Weickert. Nonlinear diffusion scale-spaces: From the continuous to the discrete setting. In *12th International Conference on Analysis and Optimization of Systems. Images, Wavelets and PDEs*, volume 219 of *Lecture Notes in Control and Information Sciences*, pages 111–118. Springer, 1996.
- [128] J. Weickert. *Anisotropic Diffusion in Image Processing*. B.G. Teubner, 1998.
- [129] J. Weickert, B. ter Haar Romeny, and M. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7:398–410, 1998.
- [130] M. Werlberger, T. Pock, M. Unger, and H. Bischof. A variational model for interactive shape prior segmentation and real-time tracking. In *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 200–211. Springer, 2009.
- [131] B. Williams, J. Spencer, K. Chen, Y. Zheng, and S. Harding. An effective variational model for simultaneous reconstruction and segmentation of blurred images. *Journal of Algorithms & Computational Technology*, 2016. submitted.
- [132] C. Wu, J. Zhang, and X. C. Tai. Augmented lagrangian method for total variation restoration with non-quadratic fidelity. *Inverse Problems and Imaging*, 5(1):237–261, 2011.

- [133] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16(2):199–209, 1997.
- [134] W. Yin. Analysis and generalizations of the linearized Bregman method. *UCLA CAM Report*, 09-42, 2009.
- [135] Y. You and M. Kaveh. A regularization approach to joint blur identification and image restoration. *IEEE Transactions on Image Processing*, 9(10):416–428, 1996.
- [136] J. Yuan, E. Bae, X. C. Tai, and Y. Boykov. A continuous max-flow approach to potts model. In *Proceedings of the 11th European Conference on Computer Vision*, volume 6316 of *Lecture Notes in Computer Science*, pages 379–392. Springer, 2010.
- [137] C. Zach, D. Gallup, J. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiphase plane sweeps. In *Vision, Modeling, and Visualization*, pages 243–252, 2008.
- [138] J. Zhang and K. Chen. Variational image registration by a total fractional-order variation model. *Journal of Computational Physics*, 293:442–461, 2015.
- [139] J. Zhang, K. Chen, and B. Yu. A 3D multigrid algorithm for the Chan-Vese model of variational image segmentation. *International Journal of Computer Mathematics*, 89:160–189, 2012.
- [140] J. Zhang, K. Chen, B. Yu, and D. Gould. A local information based variational model for selective image segmentation. *Inverse Problems and Imaging*, 8(1):293–320, 2014.
- [141] R. Zhang, X. Bresson, T. Chan, and X. C. Tai. Four color theorem and convex relaxation for image segmentation with any number of regions. *Inverse Problems and Imaging*, 7(3):1099–1113, 2013.
- [142] Y. Zhang. Recent advances in alternating direction methods: Practice and theory. In *IPAM Workshop on Continuous Optimization*, 2010.
- [143] H. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996.
- [144] Y. Zheng and K. Chen. A hierarchical algorithm for multiphase texture image segmentation. *ISRN Signal Processing*, 781653:11, 2012.