Dynamic modelling for image analysis

Nada Mansour Alfaer

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

The University of Leeds
Department of Statistics

March 2018

The candidate confirms that the work submitted is her own and that appropriate credit has been given where reference has been made to the work of others.

Abstract

Image segmentation is an important task in many image analysis applications, where it is an essential first stage before further analysis is possible. The level-set method is an implicit approach to image segmentation problems. The main advantages are that it can handle an unknown number of regions and can deal with complicated topological changes in a simple and natural way. The research presented in this thesis is motivated by the need to develop statistical methodologies for modelling image data through level sets. The fundamental idea is to combine the level-set method with statistical modelling based on the Bayesian framework to produce an attractive approach for tackling a wider range of segmentation problems in image analysis.

A complete framework for a Bayesian level set model is given to allow a wider interpretation of model components. The proposed model is described based on a Gaussian likelihood and exponential prior distributions on object area and boundary length, and an investigation of uncertainty and a sensitivity analysis are carried out. The model is then generalized using a more robust noise model and more flexible prior distributions.

A new Bayesian modelling approach to object identification is introduced. The proposed model is based on the level set method which assumes the implicit representation of the object outlines as a zero level set contour of a higher dimensional function. The Markov chain Monte Carlo (MCMC) algorithm is used to estimate the model parameters, by generating approximate samples from the posterior distribution. The proposed method is applied to simulated and real datasets.

A new temporal model is proposed in a Bayesian framework for level-set based image sequence segmentation. MCMC methods are used to explore the model and to obtain information about solution behaviour. The proposed method is applied to simulated image sequences.

Acknowledgements

All praise and glory to Allah the almighty who alone, made this objective possible to be accomplished.

I would like to thank my supervisor Dr. Robert G. Aykroyd for his continued support and guidance which helped me to reach this stage. His patience and understanding during this work was essential. His contributions to both my PhD studies, and to my experience in Leeds are greatly appreciated.

Thank you to my parents for their encouragement and support throughout my life and especially during my time in Leeds.

My thanks also go to my husband, Ahmed, and my children (Osama, Asayil, Hazim, and Dana) without whose love and encouragement, I would not have been able to complete this thesis.

Contents

Abstract ii

Acknowledgements iii

List of Figures vi

List of Tables xi

1 Introduction 1

1.1 General . . . 1

1.2 Outline of the thesis . . . 4

2 The standard level-set method 6

2.1 Introduction . . . 6

2.2 The LSM concept . . . 8

2.3 Segmentation via the LSM . . . 9

2.4 Practical implementation . . . 17

2.5 Experimental results . . . 18

2.6 Application to real data . . . 23

3 Bayesian modelling and sensitivity analysis 25

3.1 Introduction . . . 25

3.2 Definitions . . . 27

3.3 Bayesian approach . . . 27

3.4 The Gaussian-exponential model . . . 29

3.4.1 Model definition . . . 29

3.4.2 Parameter estimation . . . 30

3.5 Sensitivity analysis . . . 32

3.5.1 Prior sensitivity analysis . . . 32

3.5.2 Local influence . . . 34

3.5.3 Case influence and weighted data . . . 36

3.5.4 Missing data . . . 39

3.6 Conclusions . . . 41

4 Robust Bayesian modelling 42
4.1 Introduction 42
4.2 Choice of prior models 43
4.3 Choice of noise model 47
4.4 Simulation studies 52
4.5 Conclusions 59

5 Markov chain Monte Carlo techniques for image segmentation 60
5.1 Introduction 60
5.2 Definitions 62
5.3 Bayesian modelling 63
5.4 Markov chain Monte Carlo methods 65
5.5 Parameter estimation via the M-H algorithm 68
5.6 Implementation issues 71
5.7 Simulation experiments 75
5.8 Application to real data 96
5.9 Conclusions 101

6 Temporal modelling of the level-set matrix 102
6.1 Introduction 102
6.2 Bayesian Modelling 103
6.2.1 Likelihood 103
6.2.2 Prior models 105
6.3 Parameter estimation 106
6.4 Simulation studies 109
6.5 Conclusions 128

7 Conclusions and further work 129
7.1 Conclusions 129
7.2 Further work 130

List of Figures

2.1 *Inside area with $\phi > 0$ and outside area with $\phi < 0$.* 8

2.2 *Motion based on the time evolution. The curve $\gamma(t)$ is always given by the zero level-set of the evolving function ϕ.* 9

2.3 *The level-set based image segmentation algorithm.* 17

2.4 *Regular Heaviside and delta functions.* 18

2.5 *Synthetic images showing: (a) the true scene and (b) an example dataset.* 19

2.6 *Segmentation results showing the evolution process for the zero level-set.* 19

2.7 *Rate of convergence for $\Delta t = 1$ (in black), $\Delta t = 0.1$ (in red), and $\Delta t = 0.01$ (in green) with $\xi_{crit} = 0.1\Delta t$.* 20

2.8 *Segmentation results showing different choices of the initial zero level-set and the final zero level-set.* 21

2.9 *Segmentation results with different values of α, with $\lambda = 0$.* 22

2.10 *Segmentation results with different values of λ, with $\alpha = 0$.* 22

2.11 *Segmentation results for an image sequence showing data image and final zero level-set.* 23

2.12 *Results for selected real data showing SPECT reconstructions as a grey-level image, and final zero level-set as a curve (in red).* 24

3.1 *Posterior sensitivity with respect to changes in area prior parameter α, and $\lambda = 0$.* 33

3.2 *Posterior sensitivity with respect to changes in length prior parameter λ, and $\alpha = 0$.* 33

3.3 *Posterior sensitivity with respect to change in the area prior parameter α for the real data example.* 34

3.4 *Posterior sensitivity with respect to change in the length prior parameter λ for the real data example.* 34

3.5 *The absolute difference between the model parameters from the original data and the parameters from the perturbed data.* 35

3.6 *Grey level image showing the dataset (left), and the absolute deviations for the likelihood function, showing the influential observations (right).* 37

3.7 *Results for final zero level-set from fitting Gaussian model with unweighted data (left), zero weights for influential observations (centre), and $\omega_i = 2\hat{\sigma}_\zeta/\zeta_i$ for influential observations (right).* 38

3.8 *Segmentation results showing data images with the final zero level-set: (a-c) $\alpha = 1.2$ and $\lambda = 0$, and (d-f) $\alpha = 0$ and $\lambda = 0.1$.* 40

3.9 *Segmentation results with two missing regions for different values of the area prior α.* 40

4.1 *Examples of the beta distribution (left), and examples of the gamma distribution (right).* 44
4.2 *Examples of the Student's t-distribution.* 48
4.3 *Examples of the Poisson distribution.* 50
4.4 *The posterior density for the object area over a range of area parameters α and β.* 53
4.5 *Example of datasets with: $\mu_1 = 2$ and $\mu_0 = 3\mu_1$ (left) and $\mu_0 = 7\mu_1$ (right).* 53
4.6 *The correct classification rate (left), the SE for the area of the object (centre) and the SE for the length of the boundaries (right). The results from the Gaussian fitted model are shown with the black line, whilst the Poisson fitted model is shown with the red line.* 54
4.7 *Segmentation results of the cut-circle image with 10% of the image generated from the Gaussian distribution with different standard deviation values. The final zero level-set using the Gaussian model (first column), the final zero level-set using the weighted Gaussian model (middle column), and the final segmented image using the Student's t model (last column) are shown.* 55
4.8 *Segmentation results of the cut-circle image with 10% of the image generated from the Student's t-distribution with different scale parameter values. The final zero level-set using the Gaussian model (first column), the final zero level-set using the weighted Gaussian model (middle column), and the final segmented image using the Student's t model (last column) are shown.* 57

5.1 *The standard M-H algorithm.* 67
5.2 *The single-component random walk M-H algorithm.* 69
5.3 *Narrow-band sizes for different choices of δ.* 72
5.4 *Number of pixels inside the narrow-band region for different values of τ.* 73
5.5 *Narrow-band sizes for different values of τ.* 73
5.6 *Sample size calculation for pixels in the narrow band after burn-in period.* 75
5.7 *A sub-sample of the location of the object centre with the true location (in black) (left), the posterior mean estimate of the image (centre) and the posterior probability of pixel i being part of the object (right).* 77
5.8 *Histograms and density estimates, with 95% credible intervals for: the object mean $\hat{\mu}_*$ (left), the object area (centre) and the length of its boundary (right).* 78
5.9 *Estimation errors of the estimated boundary (left), multiple boundary traces (centre), and the estimated boundary (shown in red), with the 95% credible interval of the object boundary (shown in grey) (right).* . . 78

5.10 *Monitoring statistics for the M-H algorithm for segmenting the circular object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean* μ_**, the object area, and the length of its boundary.* 79

5.11 *Multiple zero level-set* γ showing the effect of changing the prior parameter β. The values of β are: $\beta = 0.1$ (top row), $\beta = 0.05$ (middle row), and $\beta = 0.01$ (bottom row). 81

5.12 *A sub-sample of the location of the object centre with the true location (in black) (left), the posterior mean estimate of the image (centre) and the posterior probability of pixel* i *being part of the object (right).* 82

5.13 *Histograms and density estimates, with* 90% *credible intervals for: the object mean* $\hat{\mu}_*$ *(left), the object area (centre), and the length of its boundary (right).* 83

5.14 *Estimation errors of the estimated boundary (left), multiple boundary traces (centre), and the estimated boundary (shown in red), with the* 95% *credible interval of the object boundary (shown in grey) (right).* . . 83

5.15 *Monitoring statistics for the M-H algorithm for segmenting the square object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean* μ_**, the object area, and the length of its boundary.* 84

5.16 *Sub-samples of the location of the centres with the true locations (in black) (left), the posterior mean estimate of the image (centre) and the posterior probability of pixel* i *being part of the object (right).* 85

5.17 *Histograms and density estimates, with* 90% *credible intervals for: the object mean* $\hat{\mu}_*$ *(left), the object area (centre), and the length of its boundary (right).* 86

5.18 *Estimation errors of the estimated zero level-set contour (left), multiple boundary traces (centre), and the estimated boundaries (shown in red), with the* 95% *credible interval of the boundaries (shown in grey) (right).* 87

5.19 *Monitoring statistics for the M-H algorithm for segmenting the cut-circle object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean* μ_**, the object area, and the length of its boundary.* 88

5.20 *Number of object regions across iterations.* 89

5.21 *Sub-samples of the location of the object centres with the true locations (in black) (left), the posterior mean estimate of the image (centre) and the posterior probability of pixel* i *being part of the object (right).* 89

5.22 *Histograms and density estimates, with* 90% *credible intervals for: the object mean* $\hat{\mu}_*$ *(left), the object area (centre), and the length of its boundary (right).* 90

5.23 *Estimation errors of the estimated boundaries (left), multiple boundary traces (centre), and the estimated boundaries (shown in red), with the* 95% *credible interval of the boundaries (shown in grey) (right).* 91

5.24 *Monitoring statistics for the M-H algorithm for segmenting the circular object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean μ_*, the object area, and the length of its boundary.* 92

5.25 *Sub-samples of the location of the object centres with the true locations (in black) (left), the posterior mean estimate of the image (centre), and the posterior probability of pixel i being part of the object (right).* 93

5.26 *Histograms and density estimates, with* 90% *credible intervals for: the object mean $\hat{\mu}_*$ (left), the object area (centre) and the length of its boundary (right).* 94

5.27 *Monitoring statistics for the M-H algorithm for segmenting the circular object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean μ_*, the object area, and the length of its boundary.* 95

5.28 *Estimation errors of the estimated boundaries (left), multiple boundary traces (centre), and the estimated boundaries (shown in red), with the* 95% *credible interval of boundaries (shown in grey) (right).* 96

5.29 *The image of real data showing U2OS nuclei cells.* 97

5.30 *Sub-samples of the location of the object centres (left), the posterior mean estimate of the image (centre), and the posterior probability of pixels being part of the objects (right).* 98

5.31 *Marginal posterior samples generated by the proposed method summarised using histograms and kernel density curves with* 95% *credible intervals for: the object mean $\hat{\mu}_*$ (left), the area of objects (centre), and the length of boundaries (right).* 99

5.32 *Multiple boundary traces (left), and the estimated boundaries (shown in red), with the* 95% *credible interval of the boundaries (shown in grey) (right).* 99

5.33 *Monitoring statistics for the M-H algorithm for the real data segmentation: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the mean μ_*, the area of the objects area, and the length of boundaries.* 100

6.1 *The general structure of the estimation algorithm.* 110

6.2 *Multiple boundary traces showing data images for a sequence of frames.* 111

6.3 *The posterior probability of pixel i being part of the object for all frames.* 112

6.4 *A sub-sample of the location of the object centres for all frames, with the true location (in black).* 112

6.5 *Plot of MCMC traces (left column), autocorrelation function (ACF) (middle column), and histogram with density estimates (right column) for temporal parameters in frame* 2. 114

6.6 *MCMC traces (left column), autocorrelation function (middle column), and histograms and density estimates (right column) for temporal parameters in frame* 3. 115

6.7 *MCMC traces (left column), autocorrelation function (middle column), and histograms and density estimates (right column) for temporal parameters in frame* 4. 117

6.8 *Posterior estimates with* 95% *credible intervals and true values (in black): (a) shift in the x-direction,* $\mathcal{T}_h$*, (b) shift in the y-direction,* $\mathcal{T}_v$*, (c) shift in the z-direction,* $\mathcal{T}_z$*, and (d) rotation angle,* $\mathcal{T}_r$*.* 118

6.9 *Rose diagrams for rotation angles for frames: (a)* 2*, (b)* 3*, and (c)* 4*.* 119

6.10 *Sample size calculation for the model parameters: (a) components of* $\boldsymbol{\varphi}$ *in the narrow-band region, and (b) temporal parameters.* 120

6.11 *Multiple boundary traces showing data images for a sequence of frames.* 122

6.12 *The posterior probability of pixel* i *being part of the object for all frames.* 123

6.13 *Sub-samples of the locations of the object centres for frames* 1 *to* 9 *(a) and for the two objects in frame* 10 *(b), with the true locations (in black).* 124

6.14 *Posterior estimates with the* 95% *credible intervals for the temporal parameters: (a)* $\hat{\mathcal{T}}_h$*, (b)* $\hat{\mathcal{T}}_v$*, and (c)* $\hat{\mathcal{T}}_z$*, with true values (in black).* 124

6.15 *The estimated boundaries for all frames (shown in black), with no temporal priors, with the* 95% *credible intervals for the object boundaries (shown in grey).* 125

6.16 *The estimated boundaries for all frames (shown in red), with temporal priors, with the* 95% *credible intervals for the object boundaries (shown in pink).* 126

6.17 *The SE for the estimation of: (a) area of the objects and (b) length of the boundaries. With temporal prior (in red) and with no temporal prior (in black).* 127

List of Tables

3.1 *The correct classification rate (CCR) and the squared error (SE) for the area of the object and the length of the boundaries from fitting the Gaussian model using different weights, for influential points.* 39

3.2 *The SE measures of parameters from segmentation results with two missing regions.* 41

4.1 *The CCR and the SE for the area of the object and the length of its boundaries from fitting the Gaussian model using unweighted and weighted data and the Student's t model.* 56

4.2 *The CCR and the SE for the area of the object and the length of its boundaries from fitting the Gaussian model, the weighted Gaussian model, and the Student's t model.* 58

5.1 *Summary of posterior results for circle example.* 77

5.2 *Summary of posterior results for square example.* 82

5.3 *Summary of posterior results for cut-circle example.* 86

5.4 *Summary of posterior results for three-circle example.* 90

5.5 *Summary of posterior results for L & O example.* 93

5.6 *Summary of posterior results for real data.* 98

6.1 *Summary of posterior results for frame* 2. 112

6.2 *Summary of posterior results for frame* 3. 113

6.3 *Summary of posterior results for frame* 4. 116

6.4 *Estimated correlations between parameters for frames: (a)* 2*, (b)* 3*, and (c)* 4. 119

To my family

Chapter 1

Introduction

1.1 General

In image analysis, segmentation is often an essential first task before any further analysis is carried out. The goal of image segmentation is to partition a given image into meaningful regions based on some characterising features, for example colour, intensity, or texture. These regions are often known as objects and background. Image processing can be considered as a problem in statistical inference (Besag, 1993). The key advantages of using statistical image segmentation methods is the use of statistical models which provide a flexible framework for modelling and describing the segmentation process.

In image segmentation problems, the data is a noisy copy of the true image; typically, also the data image might be blurred, which is the main focus in this thesis. Blurring has the effect of combining information from spatially close locations, where the data are similar to the true image; the regions within the image can be identified reliably using image segmentation without any additional data processing. In contrast to blurring, a projective transformation has the effect of combining information from locations that are spatially far apart (Aykroyd, 2015). Here, the image is likely to have the effects of the noise and the projective transformation as in tomography, an imaging technique which has been used widely in many medical, geological, and industrial applications (see for example Aykroyd, 2018; West et al., 2004; Aykroyd et al., 2016) for imaging an object using measurements taken outside or on the boundary of the object.

The domain of the image is discretized, and then image segmentation becomes an inverse problem. In such image problems, the segmentation task becomes more difficult or even impossible (Aykroyd, 2015).

Image segmentation approaches can be considered as low- or high-level. In low-level approaches, the image is interpreted locally based on pixel representations. The segmentation task in this approach aims to classify individual pixels using multivariate analysis techniques. In early work by Besag (1986), the image was considered as a continuous two-dimensional region divided into pixels. Each pixel has a specific colour from a finite set of colours, the true colour is unknown, and the segmentation task aims to reconstruct the image. In contrast to pixel representations, the high-level approach assumes that the image contains similar features or objects and the aim is to identify and describe the objects using statistical models. In recent years, there has been a dramatic increase in the use of Bayesian statistical models in the identification and description of the objects in images. For background to Bayesian image analysis, see Mardia and Kanji (1993), and Mardia (1994). Also for a recent review of the statistical approach to image analysis, see Aykroyd (2015) and references therein.

For Bayesian image modelling, one requires a prior model, which describes the prior information about objects in a given image, and the likelihood, which specifies the distribution of the data given information about the objects. Using Bayes' theorem, the likelihood and the prior model are combined to produce the posterior distribution of the objects, which can be used for segmentation and object recognition. However, in some practical applications, the optimization of the posterior density cannot be performed analytically, however stochastic optimization methods, the most successful of which is Markov chain Monte Carlo (MCMC), can be used in many Bayesian statistical estimation problems. Early work in statistical Bayesian imaging was carried out by Besag (1993), who suggested the use of prior information in image analysis. Then, in the work by Geman and Geman (1984), ideas about the low-level approaches were given. In high-level Bayesian image analysis, modelling could be characterized by the deformable-templates approach (see for example Grenander, 1993; Grenander and MacRae Keenan, 1993). In these methods, objects in images are represented by a deformed template (e.g a closed polygon), which is defined in terms of a number of parameters. In particular, the template can be formulated as a model for the location, size, orientation, and shape of the objects. In many problems, it is possible to separate out these parameters and to model each one independently. A simple example of a template is a circle, which can be described by parameters: the centre and the radius of the circle (Mardia, 1996). Hurn et al. (2001) represent a particular deformable template model for locating and labelling cells in microscope slides. Hobolth et al. (2003) propose a flexible continuous parametric shape model for star-shaped planar objects. Dryden et al. (2006) propose a new deformable template model based on Voronoi polygons for the segmentation of muscle fibre images.

Active curve models, also known as snakes, were first introduced by (Kass et al., 1988) as a flexible and attractive model in image segmentation. They have been used widely for image segmentation (see for example Horritt, 1999; Park et al., 2001). The technique is expressed usefully in terms of statistical terminology in Mardia (1996). In this method, the outline of an object is represented by a set of points around the boundary which are connected by splines. Deformable template techniques have also been used for Bayesian segmentation in image sequences (see for example Phillips and Smith, 1993; Mardia et al., 1992).

Similar to active contours, level-set methods (Osher and Sethian, 1988) perform segmentation by evolving a contour in an image. The contour is embedded in a higher dimensional surface where it remains closed and non-intersecting. The advantage of the level-set method compared to the snake method, is that it allows for splitting and merging of the contours without any additional processing. This presentation makes the method simple for use in image segmentation, especially in some applications where the number of regions is unknown and there is no information such as size and shape. Also the method is flexible, allowing it to be used in temporal modelling where objects move, and also change their shape and size; these changes might also include merging and splitting of the objects themselves.

The main aim of this thesis is to propose and investigate statistical methodologies for modelling image data through level sets. The fundamental idea is to combine the level-set method with statistical modelling, based on the Bayesian framework, to produce an attractive approach for tackling a wider range of segmentation problems in image analysis.

In this thesis, all calculations were performed in R statistical software (R Core Team, 2018).

1.2 Outline of the thesis

This thesis is divided into seven chapters.

Chapter 2 presents the core concepts of the level-set method, introduced by Osher and Sethian (1988). Based on the Chan-Vese model (Chan and Vese, 2001), a detailed derivation of the level-set algorithm for image segmentation is given with discussion of some practical issues. The performance of the method is investigated using simulated and real images.

Chapter 3 introduces a complete framework for the Bayesian level-set model, which allows for a wider interpretation of model components. The proposed model is described based on a Gaussian likelihood and exponential prior distributions on the area of the objects and the length of the boundaries. Various aspects of sensitivity analysis are discussed. In particular, the influence of individual measurements and re-weighting of measurements are considered, along with random perturbations of the measurements. Also, the sensitivity to the choice of prior model parameters is considered. In some applications there might be groups of data which are missing; thus the weighted likelihood approach can also be used to fit the model, as there is no requirement for a complete dataset.

In Chapter 4, a robust Bayesian model for image segmentation through level sets is investigated. The Student's t-distribution is proposed as the error model, and beta and gamma distributions are used as more flexible prior distributions to model the area of objects and the length of boundaries. The new model is used to analyse simulated image data, and compared with the fit of the model proposed in Chapter 3.

Chapter 5 describes a new Bayesian approach to level-set based image segmentation, and its implementation for the identification of objects. The Markov chain Monte Carlo algorithm is used to estimate the model parameters and to explore the proposed model, by generating approximate samples from the posterior distribution. The proposed algorithm is tested on a wide variety of grey-level simulated images and real image data.

In Chapter 6, a temporal Bayesian modelling approach based on level sets is proposed, with interest in the inclusion of temporal prior information, and the detailed methodology of how to incorporate this information is introduced and discussed. The Markov chain Monte Carlo (MCMC) method is used to perform posterior estimation, which provides measures of uncertainty, as well as point estimates. The method is illustrated using simulated sequences of images and the results are discussed.

Chapter 7 gives the final summary and conclusion of the thesis, and it outlines some suggestions for future work.

Chapter 2

The standard level-set method

2.1 Introduction

The level-set method (LSM) was introduced by Osher and Sethian in 1988 as a numerical technique to capture and analyse the motion of interfaces using appropriate partial differential equations, such as the Hamilton–Jacobi partial differential equations (Osher and Sethian, 1988). The basic idea behind the LSM is to define an $(n-1)$ dimensional interface implicitly as the zero-level of an n dimensional function called the level-set function. An appropriate choice for the level-set function ϕ is the signed distance function, see Sethian (1999) and Osher and Fedkiw (2006). The motion of the interface is controlled by the motion of the level-set function. Therefore, the LSM can handle topological changes of the interface in a natural and robust way.

Segmentation is an essential first step in many imaging problems. Segmentation involves partitioning the given image into meaningful regions, which are often an object region and a background region. For this aim, active contour models have been widely studied which use evolving contours that move towards object boundaries. In image segmentation, the idea of using active contours was introduced in 1988 by the seminal work of Kass et al. (Kass et al., 1988). In these parametric models, evolving contours are presented explicitly during the dynamic process. This approach, however, does not allow contours to split or merge; for example if the process begins with one contour then, to track multiple objects, the active contour needs to split. On the other hand, if the process begins with multiple initial contours then the active contours may need to merge. In these models the stopping criteria is based on a gradient function; thus, in noisy images where object segmentation becomes more difficult, the evolving curve might pass the object boundaries or stop at the wrong part of the image.

The LSM is a flexible and simple technique to overcome the limitations of active contour models in image segmentation. Chan and Vese (Chan and Vese, 2001) introduced a model to identify the objects in images based on the Mumford–Shah model (Mumford and Shah, 1989) and the LSM (Osher and Sethian, 1988). The Chan–Vese model is more applicable than classic segmentation methods. In this model, the boundary details are implicit and stopping the evolution process does not depend on the image gradient. This implies that the Chan–Vese model can detect objects with or without well-defined boundaries. Further, there is no need for a smoothing step before beginning the segmentation process, as noisy images can be segmented efficiently. As the model is built on the LSM formulation, objects are detected automatically.

The LSM has been used successfully in many imaging applications, including image processing, computer vision and graphics, see Tsai and Osher (2003). Recently, a survey on the development of LSMs to solve inverse problems and for optimal design was conducted by Burger and Osher (2005). Background on the LSM can be found in Sethian (1999), Osher and Fedkiw (2006) and Osher and Sethian (1988). Further, an overview of the LSM, its various applications and results can be found in Osher and Fedkiw (2001).

In the next section, the core concepts of the LSM are introduced. Section 2.3 describes the level-set approach to image segmentation based on the Chan–Vese method. Section 2.4 explains some practical issues involved in using the LSM. The method is applied to a sequence of simulated images with the numerical results given in Section 4.4. Finally, in Section 2.6, the algorithm is applied to real data.

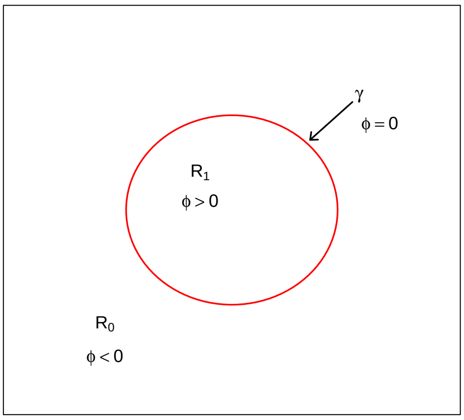


FIGURE 2.1: *Inside area with $\phi > 0$ and outside area with $\phi < 0$.*

2.2 The LSM concept

Let γ be a closed curve in the domain **S** $\subset \mathbb{R}^2$. Further assume, γ partitions **S** into non-overlapping regions; R_1, which denotes the region inside γ, and R_0, which denotes the region outside γ. The LSM is based on the implicit definition of γ as the zero-level of the embedded function $\phi(\mathbf{s})$, where $\mathbf{s} \in \mathbf{S}$. The function ϕ is called the level-set function, and it is defined for all **s**, see Figure (2.1).

The level-set function can be considered as a signed distance function, with different signs at the two sides of the boundary, $\phi(\mathbf{s}) = 0$, and $\mid \phi(\mathbf{s}) \mid$ gives the smallest distance from **s** to the boundary. For an evolving process, the zero level-set for any time t is defined as $\gamma(t) = \{\mathbf{s} \in \mathbb{R}^2 \mid \phi(\mathbf{s}, t) = 0\}$. The level-set equation describes the motion of the curve γ implicitly. Figure 2.2 shows the evolution of the zero level-set γ starting with two separate expanding curves (in red). Later in time, the topology changes, leading to a single curve. For all **s**, the function ϕ has the following properties, see Figure 2.1:

$$\begin{aligned} \phi(\mathbf{s}, t) < 0; &\quad \text{for} \quad \mathbf{s} \in R_0, \\ \phi(\mathbf{s}, t) = 0; &\quad \text{for} \quad \mathbf{s} \in \gamma, \\ \phi(\mathbf{s}, t) > 0; &\quad \text{for} \quad \mathbf{s} \in R_1. \end{aligned} \tag{2.1}$$

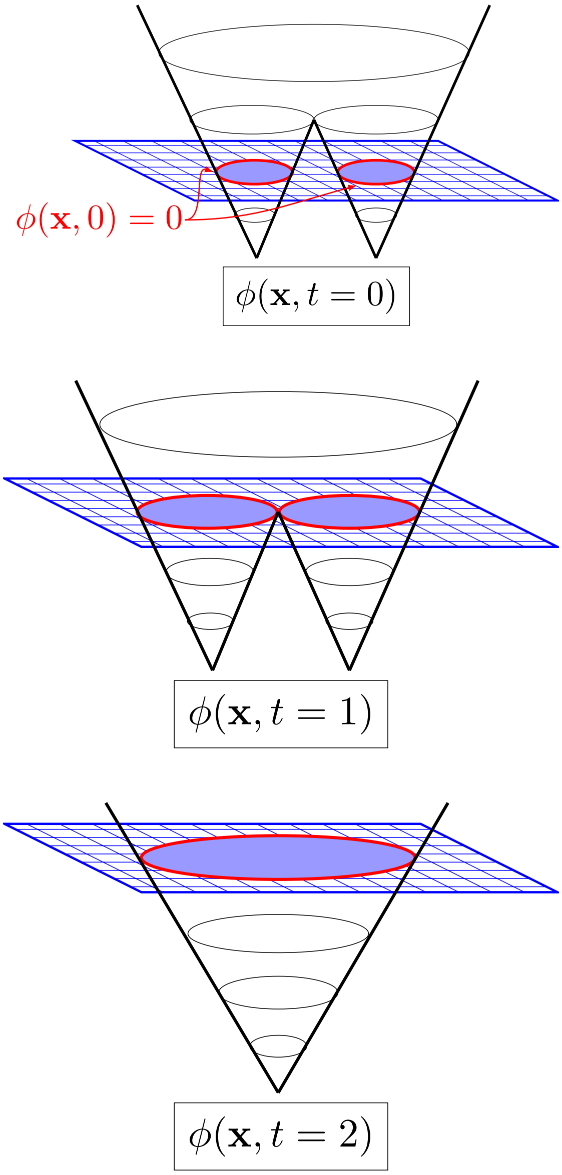


FIGURE 2.2: *Motion based on the time evolution. The curve $\gamma(t)$ is always given by the zero level-set of the evolving function ϕ.*

2.3 Segmentation via the LSM

In this section, the level set formulation of the Chan–Vese model and the details of the numerical solution are described. This model was proposed in Chan and Vese (2001) to detect objects in an image, based on the level-set method. Assume that a given function y on the domain **S**, is formed by two regions: the first region R_1 with the intensity μ_1, and the second region R_0 with the intensity μ_0. Assume further that γ is a curve, in which the region R_1 is inside γ and the region R_0 is outside γ. Consider the problem of detecting the object which is represented by the region R_1 with the intensity μ_1. The Chan–Vese solution is obtained by minimizing an energy function made up of two parts: the fitting term χ, which measures the model's mismatch, and the regularising term Υ, which ensures the stability of the solution. The first part is defined as follows:

$$\chi(y) = \int_{R_0} (y(\mathbf{s}) - \mu_0)^2 d\mathbf{s} + \int_{R_1} (y(\mathbf{s}) - \mu_1)^2 d\mathbf{s}. \tag{2.2}$$

Now, consider the Heaviside function $H(z)$, which is defined as follows:

$$H(z) = \begin{cases} 1 & \text{if } z \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

and the Dirac delta function $\delta(z)$, which is defined as follows:

$$\delta(z) = \frac{d}{dz}(H(z)) = \begin{cases} \infty & \text{if } z = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

Thus, the model mismatch can be re-written as follows:

$$\chi(\phi) = \int_{\mathbf{S}} (y(\mathbf{s}) - \mu_0)^2 (1 - H(\phi(\mathbf{s}))) d\mathbf{s} + \int_{\mathbf{S}} (y(\mathbf{s}) - \mu_1)^2 H(\phi(\mathbf{s})) d\mathbf{s}. \tag{2.5}$$

The Heaviside and Dirac delta functions are also used in the second part of the model, which consists of a weighted sum of the total area of the objects, A_*, and the total length of the boundaries of the objects, L_*. These terms can be expressed as follows:

$$A_* = A(\phi(\mathbf{s} > 0)) = \int_{\mathbf{S}} H(\phi(\mathbf{s})) d\mathbf{s}, \tag{2.6}$$

and

$$L_* = L(\phi(\mathbf{s} = 0)) = \int_{\mathbf{S}} \mid \nabla H(\phi(\mathbf{s})) \mid d\mathbf{s} = \int_{\mathbf{S}} \delta(\phi(\mathbf{s})) \mid \nabla \phi(\mathbf{s}) \mid d\mathbf{s}. \tag{2.7}$$

Thus, the regularizing term can be written as

$$\Upsilon(\phi) = \alpha \int_{\mathbf{S}} H(\phi(\mathbf{s}))d\mathbf{s} + \lambda \int_{\mathbf{S}} \delta(\phi(\mathbf{s})) \mid \nabla\phi(\mathbf{s}) \mid d\mathbf{s}, \tag{2.8}$$

where the non-negative parameters λ and α define the weight of the two components. For example, large values of α give large weight for the area of the object resulting in short boundaries, and large values of λ give large weight for the length of the boundary resulting in small objects.

Combining the parts in Equations 2.8 and 2.5, the energy function with the parameter ϕ, μ_0, μ_1 is given by

$$\begin{aligned} \mathcal{E}(\phi, \mu_0, \mu_1) \quad = \quad & \int_{\mathbf{S}} (y(\mathbf{s}) - \mu_0)^2 (1 - H(\phi(\mathbf{s})))d\mathbf{s} + \int_{\mathbf{S}} (y(\mathbf{s}) - \mu_1)^2 H(\phi(\mathbf{s}))d\mathbf{s} \\ & +\alpha \int_{\mathbf{S}} H(\phi(\mathbf{s}))d\mathbf{s} + \lambda \int_{\mathbf{S}} \delta(\phi(\mathbf{s})) \mid \nabla\phi(\mathbf{s}) \mid d\mathbf{s}. \end{aligned} \tag{2.9}$$

To simplify the notation, Equation 2.9 can be re-written again as follows:

$$\begin{aligned} \mathcal{E}(\phi, \mu_0, \mu_1) \quad = \quad & \int_{\mathbf{S}} (y(\mathbf{s}) - \mu_0)^2 (1 - H(\phi))d\mathbf{s} + \int_{\mathbf{S}} (y(\mathbf{s}) - \mu_1)^2 H(\phi))d\mathbf{s} \\ & +\alpha \int_{\mathbf{S}} H(\phi)d\mathbf{s} + \lambda \int_{\mathbf{S}} \delta(\phi)) \mid \nabla\phi \mid d\mathbf{s}. \end{aligned} \tag{2.10}$$

The parameters of the model are estimated by minimising the energy function in 2.10, with respect to ϕ, μ_0, μ_1, as follows:

$$\left(\hat{\phi}, \hat{\mu}_0, \hat{\mu}_1 \right) = \arg \min_{\phi, \mu_0, \mu_1} \mathcal{E}(\phi, \mu_0, \mu_1).$$

Hence, the estimates are found by solving the following system of equations:

$$\frac{\partial \mathcal{E}}{\partial \phi} = 0, \qquad \frac{\partial \mathcal{E}}{\partial \mu_0} = 0 \qquad \text{and} \qquad \frac{\partial \mathcal{E}}{\partial \mu_1} = 0. \tag{2.11}$$

The equations in 2.11 are obtained as follows. First, minimise the energy function 2.10 with respect to ϕ. It is not possible to compute the derivative with respect to ϕ directly; instead, the Euler–Lagrange framework and Green's theorem are used to calculate $\partial\mathcal{E}/\partial\phi$. For each integral in 2.10, the Euler–Lagrange equation is computed separately, with respect to ϕ, in the following way. First, the Euler–Lagrange equation for the integral $\int_{\mathbf{S}}(y(\mathbf{s}) - \mu_0)^2(1 - H(\phi))d\mathbf{s}$ with $g_1(\phi, \phi', t) = (y(\mathbf{s}) - \mu_0)^2(1 - H(\phi))$ is given by

$$\frac{\partial g_1}{\partial \phi} - \frac{d}{dt}\left(\frac{\partial g_1}{\partial \phi'}\right) = -(y(\mathbf{s}) - \mu_0)^2\delta(\phi) - 0 = -(y(\mathbf{s}) - \mu_0)^2\delta(\phi). \tag{2.12}$$

Secondly, the Euler–Lagrange equation for the integral $\int_{\mathbf{S}}(y(\mathbf{s}) - \mu_1)^2 H(\phi))d\mathbf{s}$ with $g_2(\phi, \phi', t) = (y(\mathbf{s}) - \mu_1)^2 H(\phi))$ is given by

$$\frac{\partial g_2}{\partial \phi} - \frac{d}{dt}\left(\frac{\partial g_2}{\partial \phi'}\right) = (y(\mathbf{s}) - \mu_1)^2\delta(\phi) - 0 = (y(\mathbf{s}) - \mu_1)^2\delta(\phi). \tag{2.13}$$

Thirdly, the Euler–Lagrange equation for the following integral: $\alpha \int_{\mathbf{S}} H(\phi)d\mathbf{s}$, with $g_3(\phi, \phi', t) = \alpha H(\phi)$ is given by

$$\frac{\partial g_3}{\partial \phi} - \frac{d}{dt}\left(\frac{\partial g_3}{\partial \phi'}\right) = \alpha\delta(\phi) - 0 = \alpha\delta(\phi). \tag{2.14}$$

Finally, the Euler–Lagrange equation for the last integral $\lambda \int_{\mathbf{S}} \delta(\phi)) \mid \nabla\phi \mid d\mathbf{s}$ with $g_4(\phi, \phi', t) = \lambda\delta(\phi)) \mid \nabla\phi \mid$ is given by

$$\begin{aligned}
\frac{\partial g_4}{\partial \phi} - \frac{d}{dt}\left(\frac{\partial g_4}{\partial \phi'}\right) &= \Big(\lambda\delta'(\phi) \mid \nabla\phi \mid\Big) - \frac{d}{dt}\Big(\lambda\delta(\phi)\nabla. \mid \nabla\phi \mid\Big), \\
&= \lambda\Big[\delta'(\phi) \mid \nabla\phi \mid -\nabla.\Big(\delta(\phi)\nabla. \mid \nabla\phi \mid\Big)\Big], \\
&= \lambda\Big[\delta'(\phi) \mid \nabla\phi \mid -\Big(\delta'(\phi)\nabla\phi.\nabla. \mid \nabla\phi \mid +\delta(\phi)\nabla(\nabla \mid \nabla\phi \mid)\Big)\Big], \\
&= \lambda\left[\delta'(\phi) \mid \nabla\phi \mid -\left(\delta'(\phi)\nabla\phi.\nabla. \mid \nabla\phi \mid +\delta(\phi)\nabla.\left(\frac{\nabla\phi}{\mid \nabla\phi \mid}\right)\right)\right], \\
&= \lambda\left[\delta'(\phi) \mid \nabla\phi \mid -\delta'(\phi) \mid \nabla\phi \mid -\delta(\phi)\nabla.\left(\frac{\nabla\phi}{\mid \nabla\phi \mid}\right)\right], \\
&= -\lambda\delta(\phi)\nabla.\left(\frac{\nabla\phi}{\mid \nabla\phi \mid}\right).
\end{aligned} \tag{2.15}$$

The total Euler–Lagrange equation for $\mathcal{E}$ is given by the summation of Equations 2.12, 2.13, 2.14 and 2.15 and is considered to be the functional derivative of $\mathcal{E}$ with respect to ϕ. Thus:

$$\frac{\partial \mathcal{E}}{\partial \phi} = -(y(\mathbf{s}) - \mu_0)^2\delta(\phi) + (y(\mathbf{s}) - \mu_1)^2\delta(\phi) + \alpha\delta(\phi) - \lambda\delta(\phi)\nabla.\left(\frac{\nabla\phi}{\mid \nabla\phi \mid}\right)$$

$$= \delta(\phi)\Bigg[- (y(\mathbf{s}) - \mu_0)^2 + (y(\mathbf{s}) - \mu_1)^2 + \alpha - \lambda\nabla.\left(\frac{\nabla\phi}{\mid \nabla\phi \mid}\right)\Bigg]. \tag{2.16}$$

The level-set equation is the negative of $\partial\mathcal{E}/\partial\phi$ in 2.16. Thus:

$$\frac{\partial \phi}{\partial t} = -\frac{\partial \mathcal{E}}{\partial \phi} = \delta(\phi)\Bigg[- (y(\mathbf{s}) - \mu_0)^2 + (y(\mathbf{s}) - \mu_1)^2 + \alpha - \lambda\text{div}\left(\frac{\nabla\phi}{\mid \nabla\phi \mid}\right)\Bigg]. \tag{2.17}$$

The equation $\partial\mathcal{E}/\partial\phi = 0$ cannot be solved explicitly, but instead it can be solved using the iteration scheme,

$$\phi^k = \phi^{k-1} + \Delta t.\frac{\partial\phi}{\partial t}\bigg|_{\phi^{k-1}}, \qquad \text{for} \quad k = 1, \ldots. \tag{2.18}$$

The estimation of ϕ is then the solution found by this iterative scheme.

Now, to calculate the estimates of μ_0 and μ_1, consider the partial derivatives of the energy function 2.10, first with respect to μ_0 and then with respect to μ_1. Keeping ϕ and μ_1 fixed and minimizing the energy function with respect to μ_0 gives us the following:

$$\frac{\partial \mathcal{E}}{\partial \mu_0} = -2\int_{\mathbf{S}}(y(\mathbf{s}) - \mu_0)(1 - H(\phi))d\mathbf{s},$$

$$= -2\Bigg[\int_{\mathbf{S}} y(\mathbf{s})d\mathbf{s} - \mu_0\int_{\mathbf{S}} d\mathbf{s} - \int_{\mathbf{S}} y(\mathbf{s})H(\phi)d\mathbf{s} + \mu_0\int_{\mathbf{S}} H(\phi)d\mathbf{s}\Bigg],$$

$$= -2\Bigg[\int_{\mathbf{S}} y(\mathbf{s})d\mathbf{s} - \mu_0\int_{\mathbf{S}}(1 - H(\phi))d\mathbf{s} - \int_{\mathbf{S}} y(\mathbf{s})H(\phi)d\mathbf{s}\Bigg]. \tag{2.19}$$

Setting Equation 2.19 to zero, the estimate of μ_0 is as follows:

$$\hat{\mu}_0 = \frac{\int_{R_0} y(\mathbf{s})d\mathbf{s}}{\int_{R_0} d\mathbf{s}}. \tag{2.20}$$

Similarly, to calculate the estimate of μ_1, with fixed ϕ and μ_0. The partial derivative of $\mathcal{E}$, with respect to μ_1, is calculated as follows:

$$\frac{\partial \mathcal{E}}{\partial \mu_1} = -2 \int_{\mathbf{S}} (y(\mathbf{s}) - \mu_1) H(\phi) d\mathbf{s},$$

$$= -2 \bigg[\int_{\mathbf{S}} y(\mathbf{s}) H(\phi) d\mathbf{s} - \mu_1 \int_{\mathbf{S}} H(\phi) d\mathbf{s} \bigg]. \tag{2.21}$$

Setting Equation 2.21 to zero, then the estimate of μ_1 is as follows:

$$\hat{\mu}_1 = \frac{\int_{R_1} y(\mathbf{s}) d\mathbf{s}}{\int_{R_1} d\mathbf{s}}. \tag{2.22}$$

For the numerical calculations, the problem needs to be discretized. Consider an $N \times N$ observed image $\boldsymbol{y} = \{y_i : i = 1, \ldots, n = N^2\}$ with pixel i located at coordinate (i_1, i_2) where $1 \leq i_1, i_2 \leq N$. The algorithm starts by considering a simple geometric shape, such as a circle, ellipse or square, as the initial zero level-set γ^0. Also the level-set function ϕ will be discretized at the pixels, this gives the level-set matrix, which is denoted by $\boldsymbol{\varphi}$. Then the initial level-set matrix $\boldsymbol{\varphi}^0$ is calculated using the signed shortest distance from a pixel to the zero level-set. First, a distance function d is defined as

$$d(i_1, i_2) = \min_{(i_1^0, i_2^0)} (|(i_1, i_2) - (i_1^0, i_2^0)|), \qquad (i_1^0, i_2^0) \in \gamma^0.$$

Then the level-set matrix is the signed distance

$$\varphi^0(i_1, i_2) = \begin{cases} -d(i_1, i_2) & \text{for all} \quad (i_1, i_2) \in R_0, \\ d(i_1, i_2) = 0 & \text{for all} \quad (i_1, i_2) \in \gamma^0, \\ +d(i_1, i_2) & \text{for all} \quad (i_1, i_2) \in R_1. \end{cases} \tag{2.23}$$

For example, if γ^0 is a circle, then the shortest distance between (i_1, i_2) and the circle is $|\sqrt{(i_1 - c_x)^2 + (i_2 - c_y)^2} - r|$, where (c_x, c_y) is the centre of the circle, and r is the radius. Hence φ^0 is calculated using the signed distance to the circle

$$\varphi^0(i_1, i_2) = \sqrt{(i_1 - c_x)^2 + (i_2 - c_y)^2} - r, \qquad \text{for all} \qquad (i_1, i_2) \in \boldsymbol{S}.$$

Then, $\hat{\mu}_0^0$ and $\hat{\mu}_1^0$ are computed using the discrete version of Equations 2.20 and 2.22, thus $\hat{\mu}_0^0$ and $\hat{\mu}_1^0$ are the averages of $\boldsymbol{y}$ in $\varphi^0 < 0$ and $\varphi^0 \geq 0$ respectively. Hence

$$\hat{\mu}_0 = \frac{1}{n_0} \sum_{(i_1,i_2)\in R_0} y(i_1,i_2), \qquad \text{and} \qquad \hat{\mu}_1 = \frac{1}{n_1} \sum_{(i_1,i_2)\in R_1} y(i_1,i_2), \tag{2.24}$$

where n_0, and n_1 are the number of pixels in regions R_0 and R_1, respectively.

In a two-dimensional space, the gradient of the function ϕ is given by $\nabla\phi = (\phi_x, \phi_y)$, where ϕ_x and ϕ_y are the first partial derivatives of ϕ with respect to the x direction and y direction, respectively. The length of the gradient is given by $\mid \nabla\phi \mid = (\phi_{xx} + \phi_{yy})^{1/2}$, where ϕ_{xx} and ϕ_{yy} are the second partial derivatives of the level-set function ϕ with respect to x and y. Thus

$$\operatorname{div}\left(\frac{\nabla\phi}{\mid \nabla\phi \mid}\right) = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{\phi_x^2 + \phi_y^2},$$

where ϕ_{xy} is the mixed partial derivative of ϕ with respect to x and then to y. To compute the level-set equation 2.17, the term div $\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ can be computed numerically using central differences for the level-set matrix $\boldsymbol{\varphi}$, which are:

$$\varphi_x(i_1,i_2) = \frac{1}{2}(\varphi(i_1+1,i_2) - \varphi(i_1-1,i_2)),$$

$$\varphi_y(i_1,i_2) = \frac{1}{2}(\varphi(i_1,i_2+1) - \varphi(i_1,i_2-1)),$$

$$\varphi_{xx}(i_1,i_2) = (\varphi(i_1+1,i_2) - \varphi(i_1,i_2)) - (\varphi(i_1,i_2) - \varphi(i_1-1,i_2)),$$

$$\varphi_{yy}(i_1,i_2) = (\varphi(i_1,i_2+1) - \varphi(i_1,i_2)) - (\varphi(i_1,i_2) - \varphi(i_1,i_2-1)),$$

and

$$\varphi_{xy}(i_1,i_2) = \frac{1}{4}(\varphi(i_1+1,i_2+1) - \varphi(i_1-1,i_2+1)) - (\varphi(i_1+1,i_2-1) - \varphi(i_1-1,i_2-1)).$$

With all components now calculated, the level-set matrix $\boldsymbol{\varphi}$ is updated with Equation 2.18. The updated level-set matrix may need to be reinitialized to the signed distance function to the zero level-set, see Sussman et al. (1994). This reinitialization step is stated as being optional in Chan and Vese (2001) but it was applied for a few numerical results presented later. The aim of this is to keep the zero level-set smooth, let $\boldsymbol{\varphi}$ be the discretization of the output ϕ from Equation 2.18, then to reinitialize $\boldsymbol{\varphi}$ and to obtain the new level-set matrix $\boldsymbol{\psi}$, the following equation is solved (Sussman et al., 1994)

$$\begin{cases} \boldsymbol{\psi}^0 = \boldsymbol{\varphi} \\ \boldsymbol{\psi} = \text{Sign}(\boldsymbol{\varphi}).(1 - \sqrt{\boldsymbol{\varphi}_x^2 + \boldsymbol{\varphi}_y^2}), \end{cases} \tag{2.25}$$

where Sign(.) is the sign function, and $\boldsymbol{\varphi}_x$ and $\boldsymbol{\varphi}_y$ are the first partial derivatives of $\boldsymbol{\varphi}$ with respect to the x direction and y direction, respectively. Equation 2.25 is numerically solved following the scheme proposed in Sussman et al. (1994). First, consider the following definitions

$$a = \varphi(i_1, i_2) - \varphi(i_1 - 1, i_2), \qquad b = \varphi(i_1 + 1, i_2) - \varphi(i_1, i_2),$$

$$c = \varphi(i_1, i_2) - \varphi(i_1, i_2 - 1), \qquad d = \varphi(i_1, i_2 + 1) - \varphi(i_1, i_2),$$

and

$$\text{Sign}(\varphi(i_1, i_2)) = \frac{\varphi(i_1, i_2)}{\sqrt{\varphi^2(i_1, i_2) + 1}}.$$

In addition, a new function $G(\varphi(i_1, i_2))$ is defined as

$$G(\varphi(i_1, i_2)) = \begin{cases} \sqrt{(\max((a^+)^2, (b^-)^2)) + \max((c^+)^2, (d^-)^2)))} - 1 \\ \qquad\qquad\qquad\qquad \text{if} \quad \varphi(i_1, i_2) > 0, \\ \sqrt{(\max((a^-)^2, (b^+)^2)) + \max((c^-)^2, (d^+)^2)))} - 1 \\ \qquad\qquad\qquad\qquad \text{if} \quad \varphi(i_1, i_2) < 0, \\ 0 \qquad \text{otherwise}, \end{cases}$$

where the $+$ superscript denotes the positive part , and the $-$ superscript denotes the negative part. Hence Equation 2.25 is updated using

$$\psi(i_1, i_2) = \varphi(i_1, i_2) - \Delta t.\text{Sign}(\varphi(i_1, i_2)).G(\varphi(i_1, i_2)). \tag{2.26}$$

The approach is stopped when the difference between $\boldsymbol{\varphi}^{k+1}$ and $\boldsymbol{\varphi}^k$ is below a tolerance, as explained in the next section. The overall algorithm is summarized in Figure 2.3.

1. Start with an initial zero level-set γ^0 and calculate the initial level-set matrix $\boldsymbol{\varphi}^0$ using Equation 2.23.
2. Compute $\hat{\mu}_0^0$ and $\hat{\mu}_1^0$ given $\hat{\boldsymbol{\varphi}}^0$ using Equation 2.24.
3. Update the level-set function $\hat{\boldsymbol{\varphi}}^k$ using Equation 2.18 to obtain $\hat{\boldsymbol{\varphi}}^{k+1}$.
4. Re-initialize and smooth $\hat{\boldsymbol{\varphi}}^{k+1}$ using Equation 2.26.
5. Check to see if the solution is stable, that is if $\xi_k < \xi_{crit}$, see Equation 2.28. Otherwise return to Step 2.

FIGURE 2.3: *The level-set based image segmentation algorithm.*

2.4 Practical implementation

As the energy function 2.10 is non-convex, the minimisation algorithm might yield many local minimisers. Thus, to obtain one global minimiser the Heaviside function $H(z)$ is numerically approximated by a function $H_\varepsilon(z)$, smoothed in the same manner as in Chan and Vese (2001) with the corresponding $\delta_\varepsilon(z)$, given by the following expressions

$$H_\varepsilon(z) = \frac{1}{2}\left[1 + \frac{2}{\pi}\text{arctan}\left(\frac{z}{\varepsilon}\right)\right] \qquad \text{and} \qquad \delta_\varepsilon(z) = \frac{\varepsilon}{\pi(\varepsilon^2 + (z)^2)}, \tag{2.27}$$

where the parameter $\varepsilon \geq 0$ controls the smoothness of the boundary. It is clear that, as $\varepsilon \to 0$, the regularized functions $H_\varepsilon(z)$ and $\delta_\varepsilon(z)$ converge to the usual functions $H(z)$ and $\delta(z)$, respectively. Figure 2.4 shows the regularized functions $H_\varepsilon(z)$ and $\delta_\varepsilon(z)$ for some values of ε.

It is not efficient to run the evolution process for a large number of iterations as this increases the computational time. However, stopping the evolution process before convergence can lead to undesirable results. It is obvious that when a steady state is attained there is no significant change in the results. This can be measured using the mean absolute change of the level-set matrix $\boldsymbol{\varphi}$ at each iteration as follows:

$$\xi_k = \frac{\sum |\boldsymbol{\varphi}^k - \boldsymbol{\varphi}^{k-1}|}{N^2}, \qquad k = 2, 3, \ldots. \tag{2.28}$$

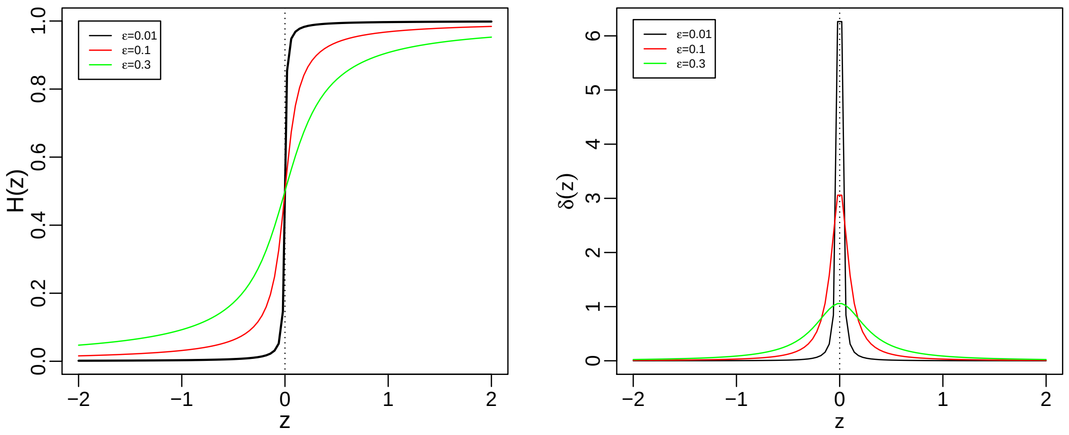


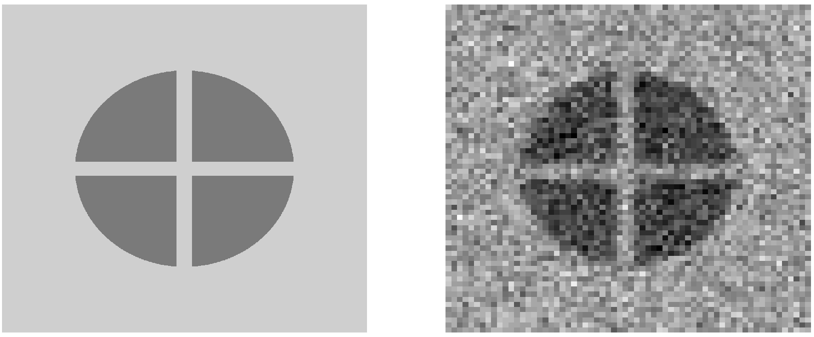
FIGURE 2.4: *Regular Heaviside and delta functions.*

For some critical value ξ_{crit}, convergence is declared when $\xi_k < \xi_{crit}$. Here, $\xi_{crit} = 10^{-2}\Delta t$, as a balance between convergence and speed. The time step Δt plays a key role in the iterative process, as it governs the speed of the motion and the stability of the solution. For satisfactory results, Δt should be chosen carefully. In Chan and Vese (2001), the time step is set as $\Delta t = 0.1$, which is appropriate for the experiments there.

2.5 Experimental results

Experiments were conducted to demonstrate the performance of the method. First, a synthetic image with a cut-circle object was considered. The parameters were set to $\lambda = 0.1$ and $\alpha = 0.4$ after some initial experiments, a sensitivity analysis will be performed later. The time step was $\Delta t = 0.01$. Here, Figure 2.5 shows a synthetic image of the region of interest. In particular, (a) shows the uncorrupted image, composed of four regions as a cut-circle shape against the background, and (b) shows the image corrupted by additive Gaussian noise with $\sigma = 1$.

Figure 2.6 shows the segmentation results for selected iterations, in particular (a) the initial zero level-set and (b–f) the evaluation of the zero level-set at later iterations. The initial zero level-set function is defined as a simple curve, that is a single circle centred in the middle. This circle was used to calculate the initial



(a) (b)

FIGURE 2.5: *Synthetic images showing: (a) the true scene and (b) an example dataset.*

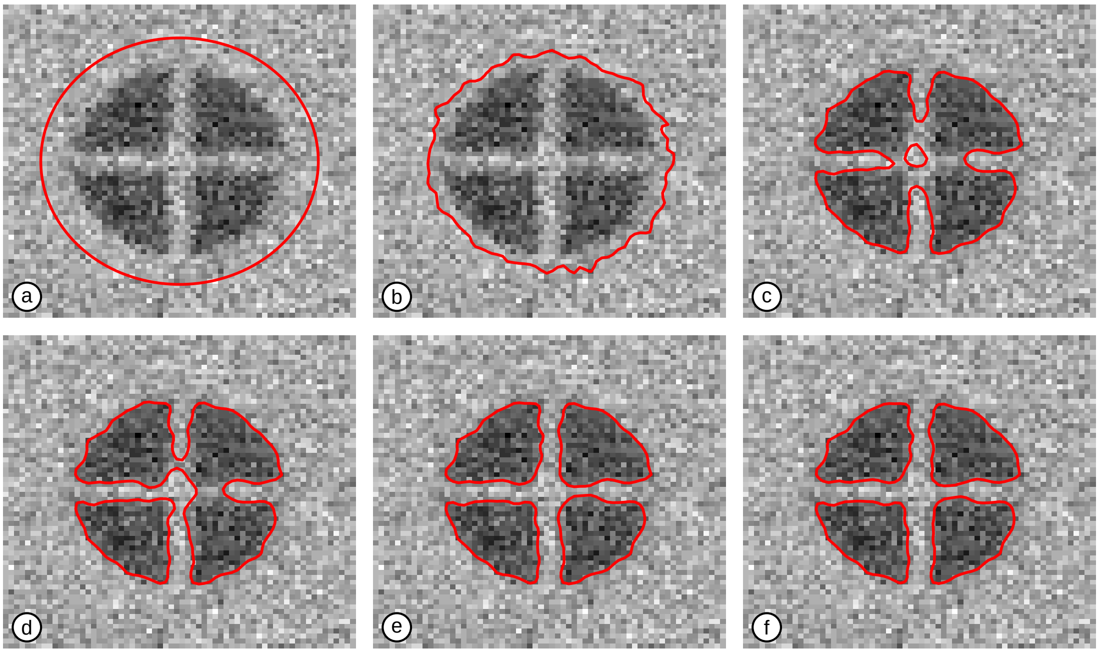


FIGURE 2.6: *Segmentation results showing the evolution process for the zero level-set.*

level-set function. As the process starts, the zero level-set begins to move towards the object boundaries. In (b), the zero level-set surrounds the object, then it comes closer to the boundaries. In (c), a hole starts to appear in the middle and grows to segment the four parts in the object separately, as shown in (d). In (e), the four parts in the cut circle are segmented successfully, with no significant change after.

The convergence of the algorithm was also investigated by calculating the rate of convergence $\xi_k/\Delta t$ for three different time step values. Figure 2.7 shows

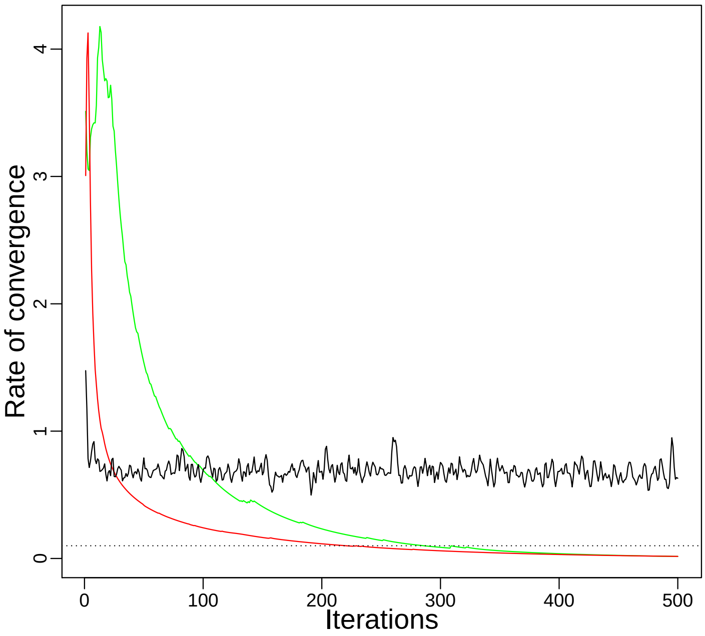


FIGURE 2.7: *Rate of convergence for* $\Delta t = 1$ *(in black),* $\Delta t = 0.1$ *(in red), and* $\Delta t = 0.01$ *(in green) with* $\xi_{crit} = 0.1\Delta t$*.*

the convergence rates for $\Delta t = 1$, $\Delta t = 0.1$ and $\Delta t = 0.01$. It is clear that, for $\Delta t = 0.1$ and $\Delta t = 0.01$, as the evolution process begins, there is a lot of change; later, these changes reduce until the zero level-set converges. However, when $\Delta t = 1$, there is no decrease in the rate of convergence; thus, the algorithm might not converge with this choice of Δt. As pictured, the algorithm stopped after around 200 iterations when $\Delta t = 0.1$ and around 300 iterations when $\Delta t = 0.01$. Thus, it can be stated that, when the time step is small, the algorithm converges slowly, but when the time step is large it may not converge at all.

The initial zero level-set can be defined arbitrarily without using the truth or the data. In order to investigate the effect of the choice of the initial zero level-set, the method was applied to the synthetic image with different initial contours. Figure 2.8 illustrates the segmentation results with several initial zero level-sets for the same synthetic image. Each row shows the initial and final zero level-sets and some in-between. The results show the flexibility of position, size and shape in the initialisation procedure. However, it is intuitive that the iterative

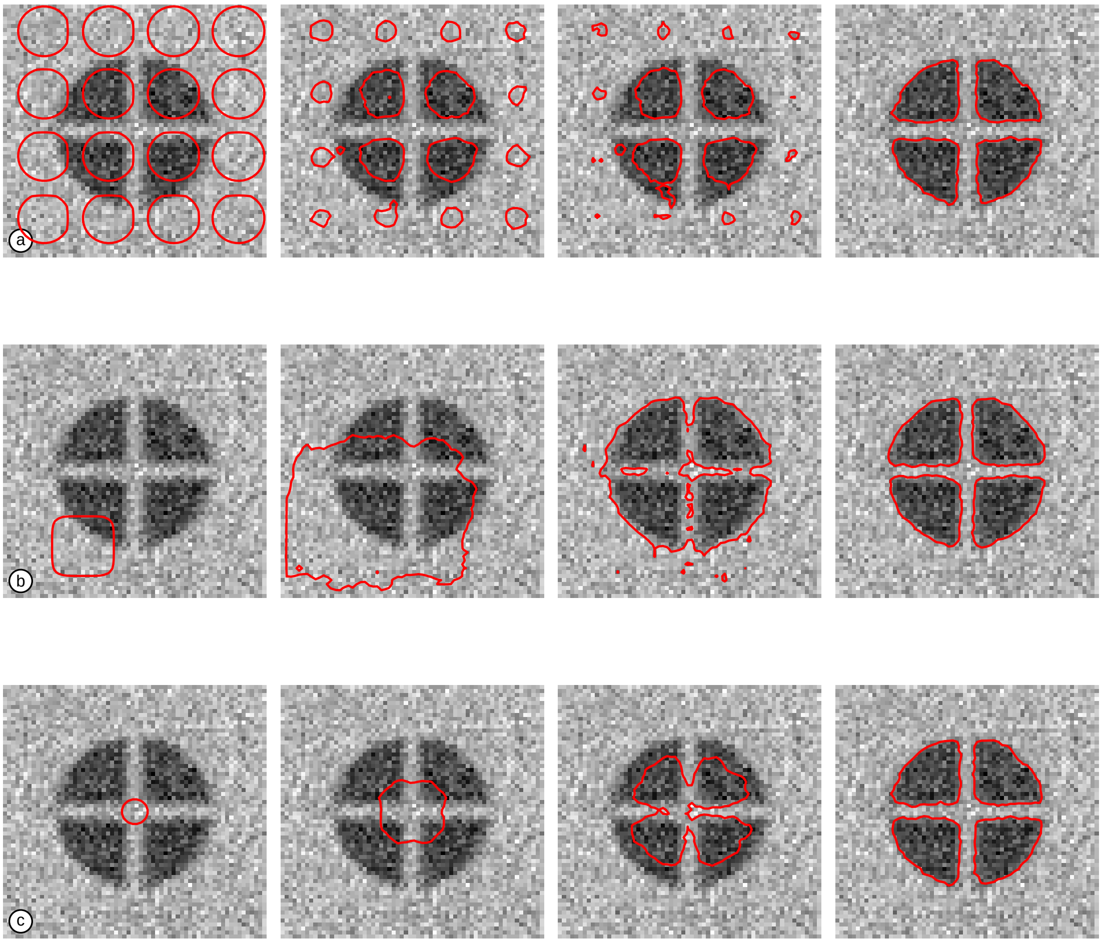


FIGURE 2.8: *Segmentation results showing different choices of the initial zero level-set and the final zero level-set.*

process is quicker and the zero level-set settles on objects quickly when the initial curve is near the true boundary.

The influence of changing the values of the regularizing weights, λ and α, in the method was also studied. Consider the synthetic image with the cut-circle object. The algorithm was applied for different values of one parameter while the other was fixed at zero. Figure 2.9 shows the final zero level-set for varying values of the length parameter λ with the area parameter $\alpha = 0$. For λ smaller than 6, all parts in the object are segmented perfectly. As λ increases, the algorithm fails to segment the cut-circle object individually. When $\lambda = 10$, the parts in the cut circle are segmented as one group, where the zero level-set is represented by a single circle. For large values of λ, the zero level-set vanishes, and the iterative process stops. Figure 2.10 shows the effect of changing the values of the area parameter α while $\lambda = 0$. For small values of α, the algorithm again fails to segment the cut circle individually, and, as α increases, the results become more desirable. For large values of α, the zero level-set starts to collapse and then disappears.

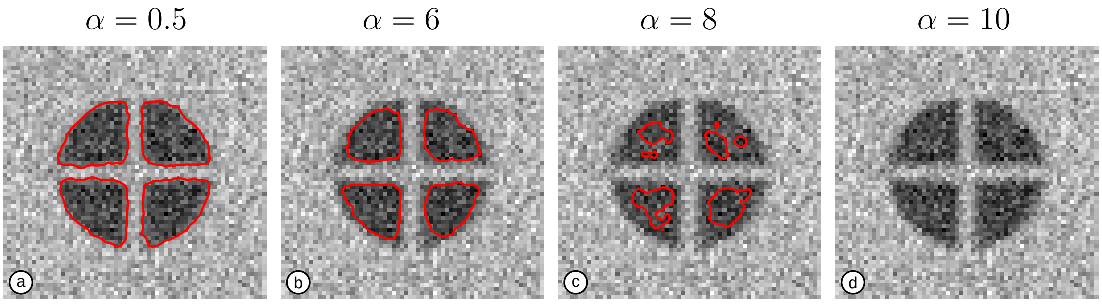


FIGURE 2.9: *Segmentation results with different values of* α*, with* $\lambda = 0$.

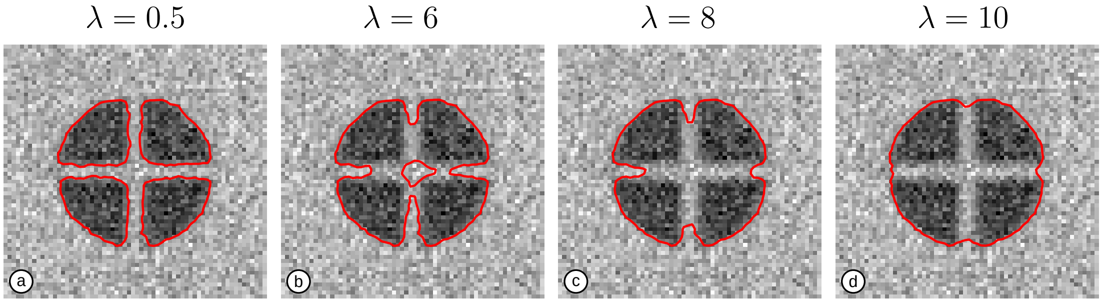


FIGURE 2.10: *Segmentation results with different values of* λ*, with* $\alpha = 0$.

The method was also applied in a simple dynamic simulation with two circles moving simultaneously. Figure 2.11 shows the evolution process, beginning with (a), when one circle hides the other circle. Later, both circles began to move and the hidden circle appears, as shown in (b), (c) and (d), until they separated, as shown in (e) and (f). The figure shows the segmentation results. It is clear the method is capable of tracking dynamically changing boundaries.

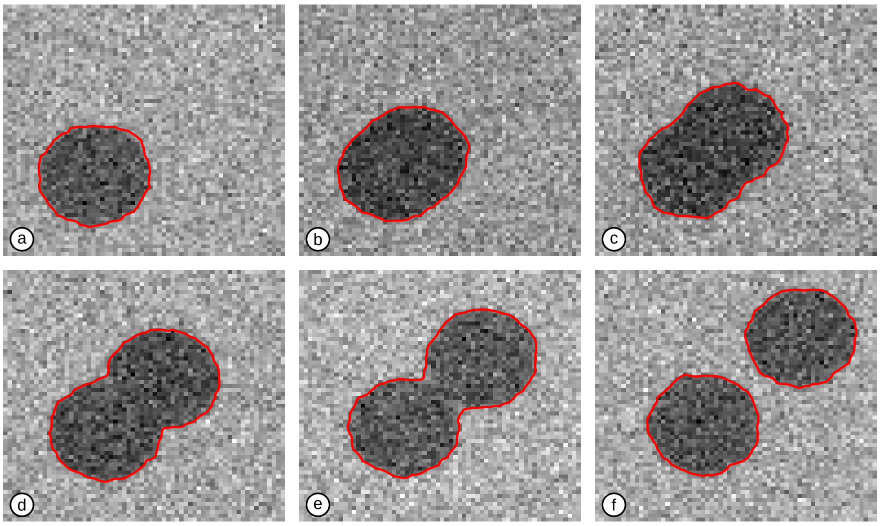


FIGURE 2.11: *Segmentation results for an image sequence showing data image and final zero level-set.*

2.6 Application to real data

The analysis of image reconstructions in Single Photon Emission Computed Tomography (SPECT) was considered as a simple illustration of application to real data. In SPECT, the patient was injected with a radioactively tagged chemical with the aim of studying metabolic rate or blood flow. Some of the emitted photons are collected in a system of detectors located around the patient using a gamma camera. Multiple projections are recorded by the gamma camera at equally spaced angles around the head of the patient. Then a three-dimensional reconstruction of the pattern of isotope concentration is formed. It is known that SPECT data is noisy, thus the reconstruction can be considered as a statistical estimation problem. However, this estimation is not easy because the coordinates where the data are collected are not those which are needed in the reconstruction (Green, 1990).

The original SPECT data was first analysed in Green (1990), but the reconstructed images were obtained from Aykroyd and Zimeras (1999). A Bayesian framework for image reconstruction from SPECT data has been proposed in Green (1990). In particular, the proposed method of reconstruction was based on a modifying Expectation Maximization (EM) algorithm with the aim of maximizing the posterior density which considers prior information about "smoothness" in the isotope concentration.

In Aykroyd and Zimeras (1999), the method was applied to produce image reconstructions of the human head, with the aim of highlighting areas of high blood flow. In particular, the images were first analysed with the assumption of spatial homogeneity, leading to over smoothing in the image reconstructions. So a fixed diameter circle was manually placed as a region of interest and the data re-analysed producing improved results. However, the results so far seen from applying the level-set method to simulated data suggest that this method could be an alternative way to define the region of interest and might allow automatic and case specific regions to be determined. Figure 2.12 shows a sequence of six cases equally spaced from the neck to the top of the head. It is clear that the level-set approach successfully identified the boundaries of the region of interest, which could be used as a basis for further analysis. The method has worked in this case, because the reconstructions are extremely smooth, and the shape of the region of interest is very simple. In particular, there is only one object which could be considered as a basic geometric object such as a circle or an ellipse, with smooth boundary.

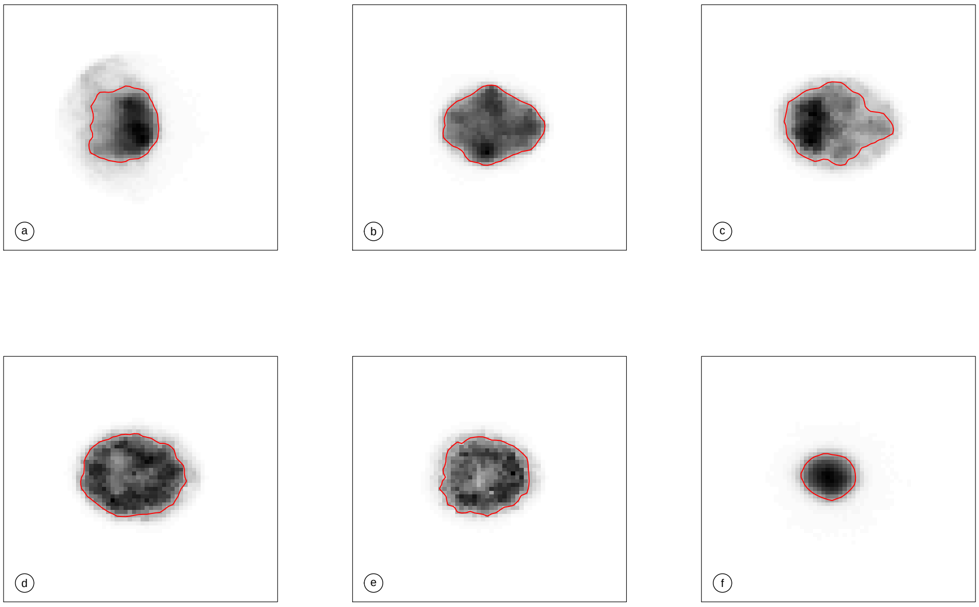


FIGURE 2.12: *Results for selected real data showing SPECT reconstructions as a grey-level image, and final zero level-set as a curve (in red).*

Chapter 3

Bayesian modelling and sensitivity analysis

3.1 Introduction

In the Bayesian approach, the relationship between data and parameters is given by a likelihood, and relationship between parameters is given by a prior distribution. Using Bayes' theorem, these components are combined to produce a posterior distribution. The idea of incorporating prior information into image analysis was first suggested by Besag (1983). Later, this idea was developed by the works of Geman and Geman (1984) and Besag (1986). In these works, low-level approaches are considered, which use pixel-based models with the goal of studying image structures pixel by pixel. However, high-level image processing, using object-based models, aims to find objects or extract geometric features in images; for example, see Besag and Green (1993). Aykroyd and Green (1991) present a study comparing these approaches; the study shows the flexibility of using a global prior against a local prior. For a comprehensive overview of statistical image analysis, the reader is directed to the collected works appearing in Mardia and Kanji (1993) and Mardia (1994).

The aim in Bayesian analysis is to fit a model to data, and to summarize the output results using a distribution on the model parameters, this gives an easy way for interpretation of many phenomena in the real world. In the standard LSM approach described in Chapter 2, the segmentation process gives only point estimates. However, using the Bayesian modelling framework will also provide information about the precision of the point estimates, that is by measuring the uncertainty in the estimates. Also, considering prior distributions for the model parameters as part of a hierarchical model will give a natural way to interpret additional information comparing to the penalty function in the Chan-Vese model. Further, Bayesian modelling gives more natural motivation for alternatively likelihood and prior models. In addition, the Bayesian model can make use of the Markov chain Monte Carlo estimation algorithm, where more flexibility in model output is possible.

The main objective of this chapter is to give a complete framework for a Bayesian level-set model. The proposed model is described based on a Gaussian likelihood and exponential prior distributions on the object area and boundary length. Sensitivity to the choice of the prior model parameters is considered; this is done to investigate the effect of changes to prior parameter values on the posterior estimates. Moreover, some influence diagnostics are proposed to assess the sensitivity of the proposed model. The influence of Gaussian perturbations of the data are considered; this is done by examining the effects on the estimation results. Then, in order to identify potentially influential observations, a leave-one-out approach is used to measure the influence of each observation on the parameter estimates. After assessing the influential observations, instead of omitting them, they are re-weighted and used in the model fitting again. In some applications, groups of the data might be missing; the weighted approach can also be used to fit the model in these situations as there is no requirement for complete data.

Section 3.2 introduces some definitions and notation that will be used later. A description of general Bayesian modelling is given in Section 3.3. Section 3.4 gives a detailed derivation of the Bayesian level-set model. Section 3.5 discusses some different aspects of sensitivity analysis; such as the effect of changes in prior parameters, the influence of individual measurements using a leave-one-out approach, and the influence of random perturbations of the measurements.

3.2 Definitions

Assume that a two-dimensional domain $\boldsymbol{S} \subset \mathbb{R}^2$ is divided into an $N \times N$ grid of equal-sized square pixels, where N is an integer. Pixel i is centred at $s_i = (i_1, i_2)$, where $1 \leq i_1, i_2 \leq N$, and the vector of these locations, $\boldsymbol{s}$, defines a finite regular square lattice. Now, define a discrete variable $x_i \in \mathbb{R}$ for each pixel. Then x_i is called the intensity, and the vector of these pixel variables, $\boldsymbol{x}$, can be shown as a grey-level image. Furthermore, assume that the domain $\boldsymbol{S}$ consists of m non-overlapping regions; for example, objects against a background. These regions are given by $\boldsymbol{R} = \{R_k : k = 0, \ldots, m-1\}$ with corresponding intensities $\boldsymbol{\mu} = \{\mu_k : k = 0, \ldots, m-1\}$. Hence, the domain $\boldsymbol{S}$ can be represented as $R_0 \cup \ldots \cup R_{m-1} = \boldsymbol{S}$ with $R_k \cap R_l = \emptyset$, for $k \neq l$, where $\emptyset$ is the empty set. Now let $R_* = \{R_1, \ldots, R_{m-1}\}$ denote all objects, and R_0 denote the background in $\boldsymbol{S}$, respectively. Thus, $R_* = R_0^c$, with corresponding intensities $\mu_* = \mu_i$ for $i = 1, \ldots, m-1$. The usual assumption is that an observed image $\boldsymbol{y} = \{y_i : i = 1, \ldots, N^2\}$ is a noisy copy of the true $\boldsymbol{x}$; this can be expressed as

$$y_i = x_i + \epsilon_i \qquad i = 1, \ldots, N^2, \tag{3.1}$$

where

$$x_i = \begin{cases} \mu_0 & \text{if} \quad s_i \in R_0, \\ \mu_* & \text{if} \quad s_i \in R_*, \end{cases} \tag{3.2}$$

and the errors ϵ_i are independent and identically distributed as $N(0, \sigma^2)$ for $i = 1, \ldots, N^2$. Therefore, the object identification problem can be addressed by estimating the number of regions, the region shapes, the intensities of the regions and the noise variance. This can be expressed as the parameter vector $\boldsymbol{\theta} = \{m, R_0, \ldots, R_{m-1}, \mu_0, \mu_*, \sigma^2\}$.

3.3 Bayesian approach

In the Bayesian approach the likelihood function and prior distribution are combined to produce the posterior distribution. The likelihood function represents the noise model, it is the conditional distribution of the data $\boldsymbol{y}$ given $\boldsymbol{\theta}$

and denoted as $f(\boldsymbol{y}|\boldsymbol{\theta})$. The data are conditionally independent and all observations have the same distribution. Thus

$$f(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{i=1}^{N^2} f(y_i|\boldsymbol{\theta}),$$

where $f(y_i|\boldsymbol{\theta})$ is the univariate distribution of y_i. If the data are continuous then $f(\boldsymbol{y}|\boldsymbol{\theta})$ is a continuous probability density function, and a popular example of this is the Gaussian distribution. Whereas, if the data are discrete then $f(\boldsymbol{y}|\boldsymbol{\theta})$ is a discrete probability mass function, and an example of this is the Poisson distribution. The Gaussian distribution is used throughout the thesis, and the Poisson distribution, which is appropriate for medical imaging applications, is studied in Chapter 4. The prior distribution expresses general beliefs about the unknown parameter vector $\boldsymbol{\theta}$. In particular, it is the distribution of the unknown parameter vector $\boldsymbol{\theta}$ and is denoted as $p(\boldsymbol{\theta})$. For each application, the choices for the prior model are different. In image segmentation problems, there are many types of additional information that can be considered in the prior distributions, such as the smoothness of the boundaries between the segmented regions, the area of the objects and the length of the boundaries, and the number of the objects.

By using Bayes' theorem the likelihood and prior distribution are combined to give a posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$, with the normalized constant $f(\boldsymbol{y})$, defined as

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{f(\boldsymbol{y})}. \tag{3.3}$$

The normalized constant $f(\boldsymbol{y})$ does not contain any information about the unknown parameters vector $\boldsymbol{\theta}$, thus it can be removed. Now, consider the log-posterior distribution

$$\log p(\boldsymbol{\theta}|\boldsymbol{y}) = \log f(\boldsymbol{y}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log f(\boldsymbol{y}). \tag{3.4}$$

In the classical estimation approach, the maximum likelihood (ML) method is used to obtain a point estimate where only the likelihood function is involved in the estimation process. In the Bayesian setting, the corresponding point estimate, obtained from the posterior distribution, is called the maximum a posteriori (MAP) estimate. However, the posterior distribution can be used in a more in depth analysis, as will be seen later. The MAP estimate is obtained by maximizing the posterior distribution in Equation 3.3, or equivalently the log-posterior distribution in Equation 3.4, thus the MAP estimate of the parameter vector $\boldsymbol{\theta}$ can be expressed as

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{y}) = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\boldsymbol{y}). \tag{3.5}$$

3.4 The Gaussian-exponential model

3.4.1 Model definition

Based on the LSM formulation in Chapter 2, and the definitions in Section 3.2, the model parameter vector is re-written to $\boldsymbol{\theta} = \{\boldsymbol{\varphi}, \mu_0, \mu_*, \sigma^2\}$, where $\boldsymbol{\varphi}$ represents the discretization version of the level-set function ϕ. Let $f(\boldsymbol{y}|\boldsymbol{\theta})$ be the conditional distribution of the observed image $\boldsymbol{y}$ given the parameter vector $\boldsymbol{\theta}$, and let the noise be modelled by a Gaussian distribution with zero mean and variance σ^2. The likelihood is then given by

$$f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma^2)^{N^2/2}} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i\in I_0}(y_i-\mu_0)^2+\sum_{i\in I_*}(y_i-\mu_*)^2\right]\right\}, \quad \sigma > 0, \tag{3.6}$$

where

$$I_0 = \{i : \varphi(\boldsymbol{s}_i) < 0\} \qquad \text{and} \qquad I_* = \{i : \varphi(\boldsymbol{s}_i) \geq 0\}.$$

The segmented object region is likely to be compact and small. Although this is an imprecise belief it must be quantified in the prior model. One option is to model the area of the object or the length of the object boundary. The simplest choice is to use an exponential distribution to describe the total area of the regions, which is denoted by A_*, or the total length of the boundaries, which is denoted by L_*. Thus, the prior distributions for A_* and L_* are given respectively as

$$p(A_*) = \alpha \exp\{-\alpha A_*\}, \qquad A_* \geq 0, \tag{3.7}$$

and

$$p(L_*) = \lambda \exp\{-\lambda L_*\}, \qquad L_* \geq 0. \tag{3.8}$$

However, it is not possible to have objects with a very big area and a very small boundary length, and so length and area variables are not independent of each other. Often, the area of the objects can give some information about the boundary length and vice versa. In this work, the area of the objects and the length of the boundaries are, however, modelled independently as a simplifying assumption. In all numerical experiments presented in Chapter 2 there are no suspicious results. In particular, there are no results where the area is smaller than expected and the length longer or vice versa which are the joint prior with higher probability than is realistic. Hence, it can be said that the assumption of independent area and length is acceptable. In addition, using this specification the prior model is easy to write down and simple to calculate. Thus A_* and L_* will be modelled using the following joint distribution

$$p(\boldsymbol{\theta}) = (\alpha\lambda)\exp\{-(\alpha A_* + \lambda L_*)\}, \qquad A_*, L_* \geq 0. \tag{3.9}$$

In a Bayesian context, substituting Equations 3.6 and 3.9 into Equation 3.3 gives the following posterior distribution

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) &= \frac{1}{(2\pi\sigma^2)^{N^2/2}} \exp\left\{ -\frac{1}{2\sigma^2}\left[\sum_{i\in I_0}(y_i-\mu_0)^2 + \sum_{i\in I_*}(y_i-\mu_*)^2\right]\right\} \\ &\qquad \times(\alpha\lambda)\exp\{-(\alpha A_* + \lambda L_*)\}, \\ &= \frac{\alpha\lambda}{(2\pi\sigma^2)^{N^2/2}} \exp\left\{ -\frac{1}{2\sigma^2}\left[\sum_{i\in I_0}(y_i-\mu_0)^2 + \sum_{i\in I_*}(y_i-\mu_*)^2\right] \right. \\ &\qquad \left. -(\alpha A_* + \lambda L_*)\right\}. \end{aligned} \tag{3.10}$$

This will be used as the basis for estimation.

3.4.2 Parameter estimation

The log-posterior distribution corresponding to 3.10 is

$$\begin{aligned} L(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}|\boldsymbol{y}) &= -\frac{N^2}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\left[\sum_{i\in I_0}(y_i-\mu_0)^2 + \sum_{i\in I_*}(y_i-\mu_*)^2\right] \\ &\qquad -\alpha A_* - \lambda L_* + C, \end{aligned} \tag{3.11}$$

where C contains all constants in Equation 3.10. This equation is the discrete equivalent of the negative of the level-set energy function in the standard LSM. Similar to maximum likelihood estimation in classical statistics, in a Bayesian context a point estimate can be found as the value which corresponds to the maximum of the posterior distribution, this is called the maximum a posteriori (MAP) estimate

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{y}),$$

or equivalently based on the log-posterior

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\boldsymbol{y}).$$

If the maximum is located at a turning point then $\hat{\boldsymbol{\theta}}_{MAP}$ can be found by differentiation. This gives the following equation system, which must be solved simultaneously

$$\frac{\partial L}{\partial \boldsymbol{\varphi}} = 0, \qquad \frac{\partial L}{\partial \mu_0} = 0, \qquad \frac{\partial L}{\partial \mu_*} = 0, \qquad \frac{\partial L}{\partial \sigma^2} = 0.$$

This equation system is solved as follows: first, to estimate $\boldsymbol{\varphi}$, fix the parameters μ_0, μ_1 and σ^2. The partial derivative of 3.11 with respect to $\boldsymbol{\varphi}$ is given by, as has already been shown in Chapter 2,

$$\frac{\partial L}{\partial \boldsymbol{\varphi}} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \boldsymbol{\varphi}} = \delta(\boldsymbol{\varphi})\left[-(\boldsymbol{y}-\mu_0)^2+(\boldsymbol{y}-\mu_*)^2+\alpha-\lambda \text{div}\left(\frac{\nabla\boldsymbol{\varphi}}{|\nabla \boldsymbol{\varphi}|}\right)\right]. \tag{3.12}$$

Then, the corresponding updating equation is

$$\boldsymbol{\varphi}^k = \boldsymbol{\varphi}^{k-1} + \Delta t \cdot \left.\frac{\partial(\boldsymbol{\varphi})}{\partial t}\right|_{\boldsymbol{\varphi}^{k-1}}, \qquad \text{for} \quad k = 1, ... \tag{3.13}$$

Second, to estimate the means μ_0, μ_* and the variance σ^2, consider the following partial derivatives

$$\frac{\partial L}{\partial \mu_0} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \mu_0} = \frac{1}{\sigma^2}\sum_{i\in I_0}(y_i - \mu_0),$$

$$\frac{\partial L}{\partial \mu_*} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \mu_*} = \frac{1}{\sigma^2}\sum_{i\in I_*}(y_i - \mu_*),$$

and

$$\frac{\partial L}{\partial \sigma^2} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \sigma^2} = -\frac{N^2}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\left[\sum_{i\in I_0}(y_i-\mu_0)^2 + \sum_{i\in I_*}(y_i-\mu_*)^2\right].$$

The parameter values that maximise $\log p(\boldsymbol{\theta}|\boldsymbol{y})$ are obtained by analytically solving the following equations

$$\frac{\partial L}{\partial \mu_0} = 0, \quad \frac{\partial L}{\partial \mu_*} = 0, \quad \text{and} \quad \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \sigma^2} = 0.$$

Therefore, the parameter estimates are given by

$$\begin{aligned} \hat{\mu}_0 &= \frac{1}{n_0}\sum_{i\in I_0} y_i, \qquad \hat{\mu}_* = \frac{1}{n_*}\sum_{i\in I_*} y_i, \qquad \text{and} \\ \hat{\sigma}^2 &= \frac{1}{N^2}\left[\sum_{i\in I_0}(y_i-\mu_0)^2 + \sum_{i\in I_*}(y_i-\mu_*)^2\right], \end{aligned} \tag{3.14}$$

where n_0 and n_* are the number of data points in regions I_0 and I_* respectively, and so $N^2 = n_0 + n_*$.

3.5 Sensitivity analysis

3.5.1 Prior sensitivity analysis

The influence of changing the prior parameter values is studied; this is based on the Gaussian-exponential model. In particular, the method is applied first to the simulated data shown in Figure 2.11 (d), and second to the real data in Figure 2.12 (b) for different choices of α and λ.

First, the results based on simulated data are discussed. Figure 3.1 shows the effect of changing the area prior parameter α while the length prior parameter λ is fixed at zero. As the area prior α increases, both the length and area posterior estimates gently decrease. However, the change in the length posterior is more gradual than the change in the area posterior. When α is 6.5, the two circles separate, and then the circles start to shrink until they disappear. However, it can be said that the estimation is robust to changes in the area prior parameter

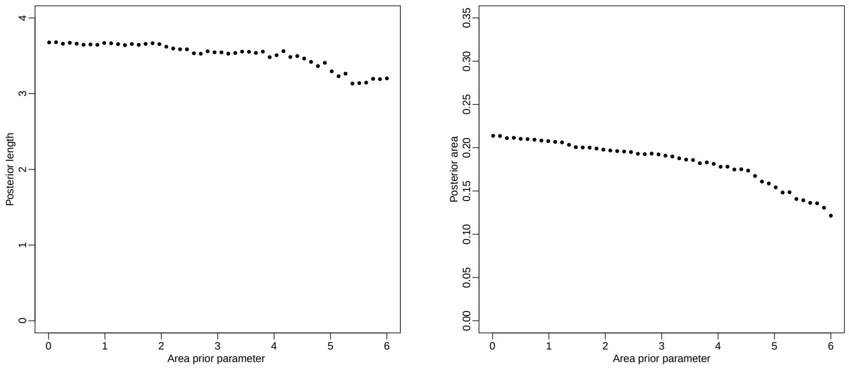


FIGURE 3.1: *Posterior sensitivity with respect to changes in area prior parameter* α*, and* $\lambda = 0$.

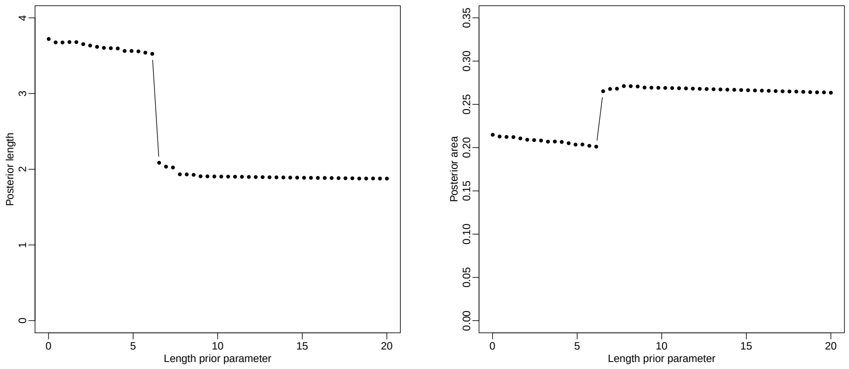


FIGURE 3.2: *Posterior sensitivity with respect to changes in length prior parameter* λ*, and* $\alpha = 0$.

over a broad range of values.

Similarly, Figure 3.2 shows the effect of changing the length prior parameter λ while the area prior parameter α is fixed at zero. It is clear from the figure that, as the value of λ increases, there is a slight change in both length and area. However, when $\lambda = 5$, there is a significant change in the length and area; then there is a slight decrease. Shortly after, the estimated values become zero. Therefore, for small values of λ, the true objects are segmented very well. Later, they are both joined with small changes until $\lambda = 5$, when the circles start to disappear. Thus, the estimation results are the same over a wider range of values compared to the area prior parameter.

Second, the sensitivity analysis for the real data in Figure 2.12 (b) is discussed.

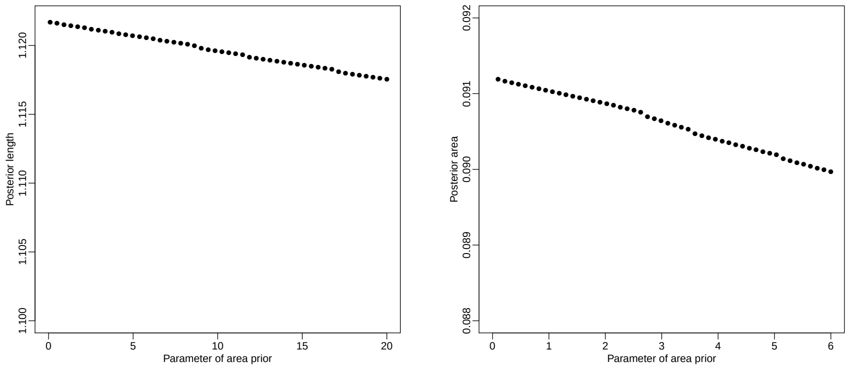


FIGURE 3.3: *Posterior sensitivity with respect to change in the area prior parameter α for the real data example.*

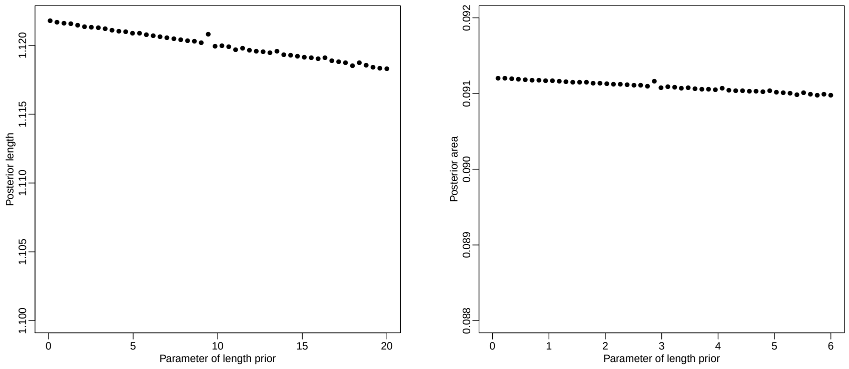


FIGURE 3.4: *Posterior sensitivity with respect to change in the length prior parameter λ for the real data example.*

Figure 3.3 shows a stable decrease in the posterior estimates for the range of α considered. In Figure 3.4, both posterior estimates for the area and the length of the object decrease gradually over the range of λ considered. Therefore, the results here show the robustness of the proposed algorithm over a range of values of length and area prior parameters; they also show that the output of the analysis is not sensitive to the precise choice in the prior parameter values.

3.5.2 Local influence

The aim of this section is to study the effect of modest changes in the data on the parameter estimates. In particular, all data are perturbed by random noise;

then, reapplying the analysis, the results are used to measure the local influence of these perturbations. The local influence method was proposed by Cook (1986) as a method for identifying influential observations in regression analysis. The method has become a popular diagnostic tool because it can be applied to any model.

Consider a given data set $\boldsymbol{y}$ that is perturbed by defining a vector $\boldsymbol{\omega} = \{\omega_i\colon 1 \leq i \leq N^2\}$, where the ω_i are independently distributed from a Gaussian distribution with mean zero and variance σ_ω^2. Therefore, the perturbed data is given by

$$\boldsymbol{y}_{\boldsymbol{\omega}} = \boldsymbol{y} + \boldsymbol{\omega}. \tag{3.15}$$

Also, there is a non-perturbation vector $\boldsymbol{\omega}_0$ such that $\boldsymbol{y}_{\boldsymbol{\omega}_0} = \boldsymbol{y}$.

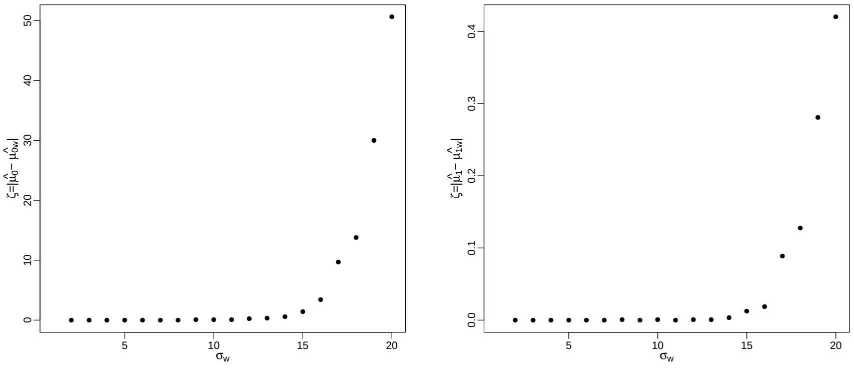


FIGURE 3.5: *The absolute difference between the model parameters from the original data and the parameters from the perturbed data.*

Here, this local influence is assessed by comparing the parameter estimates from the original and perturbed models as $\sigma_{\boldsymbol{\omega}}^2$ increases. Figure 3.5 (a) shows the absolute difference between $\hat{\mu}_0$ in the original data and $\hat{\mu}_0$ in the perturbed data, that can be defined as $\zeta = |\hat{\mu}_0 - \hat{\mu}_{0\omega}|$. In the same figure, (b) shows the absolute difference between $\hat{\mu}_1$ in the original data and $\hat{\mu}_1$ in the perturbed data, that can be defined as $\zeta = |\hat{\mu}_1 - \hat{\mu}_{1\omega}|$. It is clear in both panels that, as the noise level increases, there is an increase in the absolute differences for the parameters $\hat{\mu}_0$ and $\hat{\mu}_1$, showing that the accuracy of the segmentation results is affected, but the effect of the perturbations is gradual and smooth as $\sigma_{\boldsymbol{\omega}}^2$ changes.

3.5.3 Case influence and weighted data

For a given data set, not all observations have the same impact on the segmentation results. The aim here is to identify data points that have a significant impact on the estimation results; these points are called influential points. In particular, a leave-one-out approach is used to study the influence of individual observations on the final results. Consider that $p(\boldsymbol{\theta}|\boldsymbol{y})$ is the log-posterior for the parameter vector $\boldsymbol{\theta}$ and the data $\boldsymbol{y} = \{y_1, ..., y_{N^2}\}$. The approach used here is based on repeatedly omitting one observation at a time and doing an analysis on the remaining observations. As a result, any significant change in the results suggests that the omitted observation is influential. The data omitting the i^{th} observation is denoted by $\boldsymbol{y}_{-i} = \{y_1, ...y_{i-1}, y_{i+1}, ..., y_{N^2}\}$ with corresponding $\log p(\boldsymbol{\theta}|\boldsymbol{y}_{-i})$, and the MAP estimate is denoted by $\hat{\boldsymbol{\theta}}_{-i}$.

Let $\hat{\boldsymbol{\theta}}$ be a parameter estimate using all data and $\hat{\boldsymbol{\theta}}_{-i}$ be a parameter estimate using all data while omitting the i^{th} observation. Then, the absolute deviation from the likelihood function $L(\hat{\boldsymbol{\theta}})$ is given by

$$\zeta_i = |L(\hat{\boldsymbol{\theta}}) - L(\hat{\boldsymbol{\theta}}_{-i})|. \tag{3.16}$$

As influential observation i tends to have a high value of ζ_i; when ζ_i is approximately zero, observation i has no influence on the parameter estimates. To assess the influence of these points, ζ_i is calculated for each observation to give $\boldsymbol{\zeta} = \{\zeta_1, \dots, \zeta_{N^2}\}$. For each ζ_i, a confidence interval is then constructed using the sample median absolute deviation (MAD) as a robust estimate of the standard deviation of $\boldsymbol{\zeta}$, which is given by

$$\hat{\sigma}_{\boldsymbol{\zeta}} = 1.4826 \cdot \text{median}(|\zeta_1|, \dots, |\zeta_{N^2}|). \tag{3.17}$$

All points with $\zeta_i > 2\hat{\sigma}_{\boldsymbol{\zeta}}$ are identified as influential points.

In the case where some influential points are identified, observations should not be treated equally. The observations with a large influence will be given less weight, and the observations with a small influence will be given more

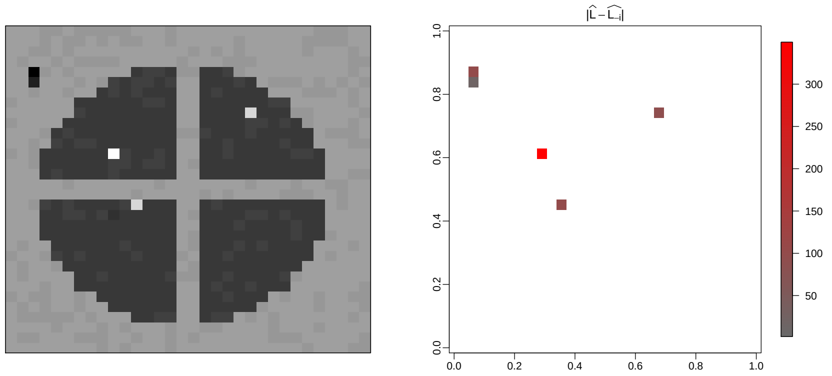


FIGURE 3.6: *Grey level image showing the dataset (left), and the absolute deviations for the likelihood function, showing the influential observations (right).*

weight. Equivalently, a weighted log-likelihood can be defined as

$$\log f(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{\omega}) = -\frac{N^2}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\left[\sum_{i\in I_0}\omega_i(y_i-\mu_0)^2 + \sum_{i\in I_*}\omega_i(y_i-\mu_*)^2\right], \quad (3.18)$$

where $\omega_i \geq 0$ is the weight for the i^{th} observation. Two cases can be considered; first if the influential observations are known, then $\omega_i = 0$ for the influential observations and $\omega_i = 1$ for non-influential observations. Second, when the influential observations are unknown, a leave-one-out approach is used to identify potentially influential data points. Then, weight $\omega_i = 2\hat{\sigma}_\zeta/\zeta_i$ is assigned to each influential observation. Equation 3.18 can then be used to re-fit the model again, and the corresponding estimate $\hat{\boldsymbol{\theta}}_{\boldsymbol{\omega}}$ is obtained.

Now the behaviour of influential observations is investigated. Consider Figure 3.6(left) which shows an example of simulated data. The data was simulated from the Gaussian distribution with $\mu_0 = 10$, $\mu_1 = 5$ and standard deviation $\sigma = 0.1$. Five observations are chosen randomly, and their intensity values are changed. In particular, three object observations were replaced by $\mu_i = 10$ to be from the background region and two background observations were replaced by $\mu_i = 5$ to be from the object region. A leave-one-out approach is used to identify potentially influential observations. The absolute deviations of the likelihood function were calculated, and a threshold was set at double the estimated standard deviation of all the absolute deviations. The results are shown in Figure 3.6(right), it is clear that the influential points are identified as illustrated by the red colour (strong influence) and the grey colour (weak influence).

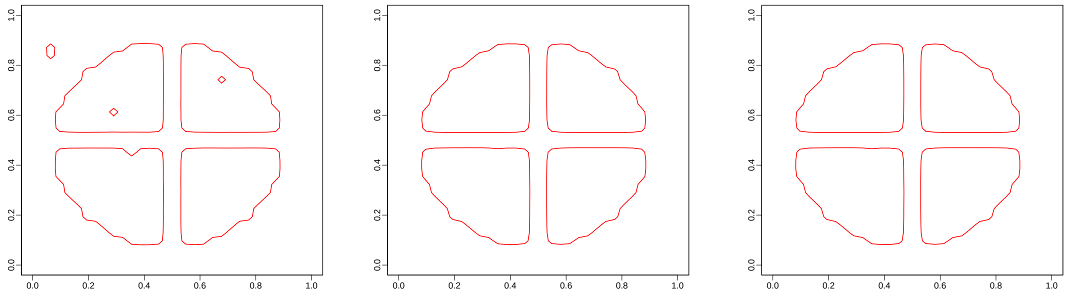


FIGURE 3.7: *Results for final zero level-set from fitting Gaussian model with unweighted data (left), zero weights for influential observations (centre), and* $\omega_i = 2\hat{\sigma}_{\zeta}/\zeta_i$ *for influential observations (right).*

The data has first been analysed using the Gaussian model with unweighted data, with the result for the final zero level-set shown in Figure 3.7(left). It is clear that the influential points were classified wrongly, the background observations were segmented to be from the object region, and the object observations were segmented to be from the background region. The weighted likelihood defined in 3.18 is now used to fit the model. Two cases are considered: first zero weights are assigned to the influential observations and the Gaussian model was fitted, the result is shown in Figure 3.7(centre). Then the weights $\omega_i = 2\hat{\sigma}_{\zeta}/\zeta_i$ are assigned for the influential observations, and again the model was fitted, with the result as shown in Figure 3.7(right). It is clear that by fitting the Gaussian model with weighted influential observations, the four parts of the cut-circle were identified with no misclassified pixels. The influential observations were segmented correctly to be in their original regions. To summarise the results two output measures will be considered: first the squared error (SE) for the area of the object and the length of the boundaries, which is defined as squared difference between the estimated value and the true value, and the correct classification rate (CCR), which is defines as the number of correctly classified pixels divided by the total number of pixels. Table 3.1 displays the numerical results for the SE and the CCR. It is clear that fitting the Gaussian model with weighted influential observations gives better results, with low SE and high CCR.

Weights	CCR	SE for the area	SE for the length
Unweighted	0.995	0.000003	0.05
$\omega_i = 0$	1.000	0.000001	0.03
$\omega_i = 2\hat{\sigma}_{\boldsymbol{\zeta}}/\zeta_i$	1.000	0.000001	0.03

TABLE 3.1: *The correct classification rate (CCR) and the squared error (SE) for the area of the object and the length of the boundaries from fitting the Gaussian model using different weights, for influential points.*

3.5.4 Missing data

Images with missing data frequently occur in practical applications. In image analysis, various methods have been proposed to handle this problem. One approach is to interpolate the values of these missing data using some estimation method, where the missing data are treated as unknown parameters (Little and Rubin, 2014). However, the proposed algorithm in this chapter can deal with this problem naturally. In particular, the LSM is used to identify objects in a given image through the iterative estimation procedure. In calculating the level-set matrix $\boldsymbol{\varphi}$ only the pixel positions are used, the values of the data are not needed, and so φ can be defined with missing data. Therefore, the log-likelihood function in the Gaussian-exponential model does not require a complete dataset. Overall, the level-set value can be defined for those missing data, and there is no need for any fill-in techniques. In addition to the segmentation results, this algorithm allows for the segmentation of those missing values.

The following example demonstrates the capacity of the proposed algorithm for detecting regions with missing data and how it can segment the image automatically. Figure 3.8 shows data images with different missing regions; white pixels show the locations of these missing data. The method is applied to the data images with prior parameters $\alpha = 0$, and $\lambda = 0.1$, then it is applied with prior parameters $\alpha = 1.2$, and $\lambda = 0$. The results of the final zero level-set are shown in the figure. It is clear that for $\alpha = 0$, and $\lambda = 0.1$, the objects with missing regions are identified well. However, with prior parameters $\alpha = 1.2$, and $\lambda = 0$, the objects were identified but the missing regions were segmented

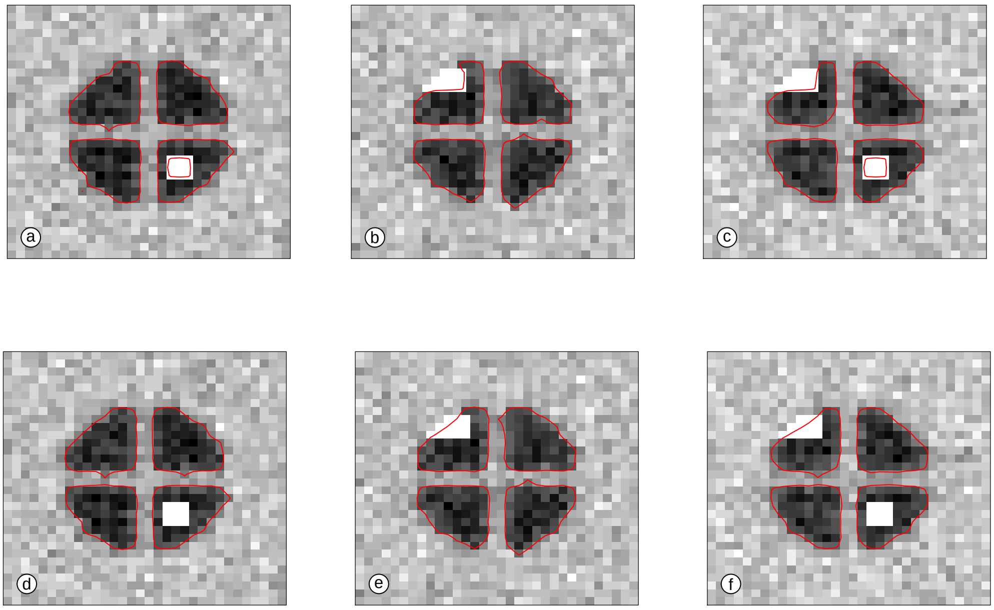


FIGURE 3.8: *Segmentation results showing data images with the final zero level-set: (a-c) $\alpha = 1.2$ and $\lambda = 0$, and (d-f) $\alpha = 0$ and $\lambda = 0.1$.*

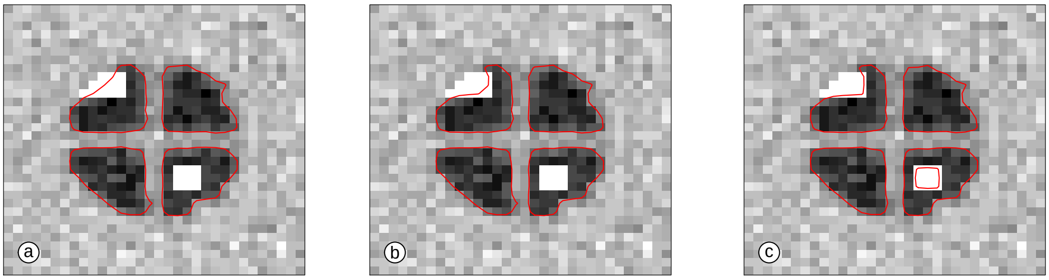


FIGURE 3.9: *Segmentation results with two missing regions for different values of the area prior α.*

wrongly.

To investigate how the segmentation results are influenced by changing prior parameters, the prior for the area is considered. Figure 3.9 shows the final zero level-set for different values of the prior parameter α. It is clear that, as α increases, those missing regions are identified as background. This is due to the aim of minimizing the area. The squared error is calculated for the parameter estimates; with the results shown in Table 3.2. It is clear that the prior parameter α affects the segmentation results in this situation.

	μ_0	μ_1	σ^2	*Length*	*Area*
$\alpha = 0.1$	0.3700	0.0147	0.0288	0.1308	0.0000
$\alpha = 1.0$	0.3178	0.0175	0.0300	0.1364	0.0003
$\alpha = 2.4$	0.2065	0.0260	0.0375	0.0398	0.0014

TABLE 3.2: *The SE measures of parameters from segmentation results with two missing regions.*

3.6 Conclusions

In this chapter, a Bayesian re-interpretation of the standard LSM was given. The estimation algorithm was given in full details to obtain the MAP estimate of the level-set matrix. The method has been successfully applied to simulated images, as well as to real 2D images obtained from medical SPECT data. A sensitivity analysis was performed where the effect of changes in prior parameters were considered and the influence of individual measurements along with random perturbations of the measurements were studied. A further extension, using a weight function, was proposed to account for unreliable data values or even missing data. The corresponding model was applied to simulated data showing its suitability for image segmentation.

Chapter 4

Robust Bayesian modelling

4.1 Introduction

The aim of robust analysis is to build models that are not sensitive to small or moderate changes in the modelling assumptions. In the Gaussian-exponential model, the prior distributions are exponential distributions; however, with this assumption, the most likely values are zero. This means that for large values of the prior parameters, the evolving curve collapses. This suggests the use of more appropriate distributions that allow for a non-zero mode. Here, beta and gamma distributions are used as more flexible prior distributions to model the area of the object and the length of the boundaries, respectively. In the Gaussian-exponential model the data model is a Gaussian distribution. To allow more dispersion in the data, especially when the data contains outliers, one suggestion is to replace the Gaussian distribution with the Student's t-distribution. The Student's t-distribution is used to replace a Gaussian distribution in robust linear regression; see, for example, Gelman et al. (2014), and in robust mixture modelling; see, for example, Peel and McLachlan (2000). With discrete data an appropriate choice for a robust model is to replace a Gaussian distribution with a Poisson distribution (Weir and Green, 1994). In this chapter, the Student's t-distribution is proposed as the data model as a robust alternative to the Gaussian distribution in the Gaussian-exponential model proposed in Chapter 3. For more details of Bayesian robustness analysis and developments of the influence diagnostic methods, see Berger et al. (1994), Insua and Ruggeri (2012) and the references there.

Section 4.2 introduces more flexible generalizations of prior distributions for object area and boundary length. Some alternatives to the Gaussian likelihood distribution in the Gaussian-exponential model are proposed in Section 4.3. Finally, detailed simulation studies are reported in Section 4.4.

4.2 Choice of prior models

In Bayesian modelling, the prior distribution presents expert knowledge or general beliefs about unknown parameters. In the Gaussian-exponential model, the exponential distribution was used as the prior distributions to describe the area of the objects and the length of their boundaries; where the small and compact regions are more likely than large or irregular regions. This choice, however, has the unusual property that the most likely values are close to zero, which is not a meaningful statement; and in fact this might be the cause of the rapid disappearance of the objects for large values of the prior parameters. Instead it might be more appropriate to use a distribution allowing a non-zero mode, such as a gamma distribution. Also, especially for area the distribution should have a finite upper limit; as the total area cannot be greater than the whole domain size which can be suitably accommodated using, for example, a beta distribution.

Let A_* be the total area of the objects and A_{max} be the area of the image. Then the beta distribution with parameters α and β is used to model the proportion of area of the objects, hence the corresponding prior distribution is

$$p(A_*) = \frac{1}{B(\alpha,\beta)}(A_*/A_{max})^{\alpha-1}(1-A_*/A_{max})^{\beta-1}, \qquad 0 \le A_* \le A_{max}, \tag{4.1}$$

where $B(.,.)$ is the beta function. If $A_*/A_{max} \sim$ Beta(α,β), then some properties of the Beta distribution are

$$(i) \quad E\left(\frac{A_*}{A_{max}}\right) = \frac{\alpha}{(\alpha+\beta)},$$

$$(ii) \quad \text{var}\left(\frac{A_*}{A_{max}}\right) = \frac{\alpha\beta}{[(\alpha+\beta)^2(\alpha+\beta+1)]},$$

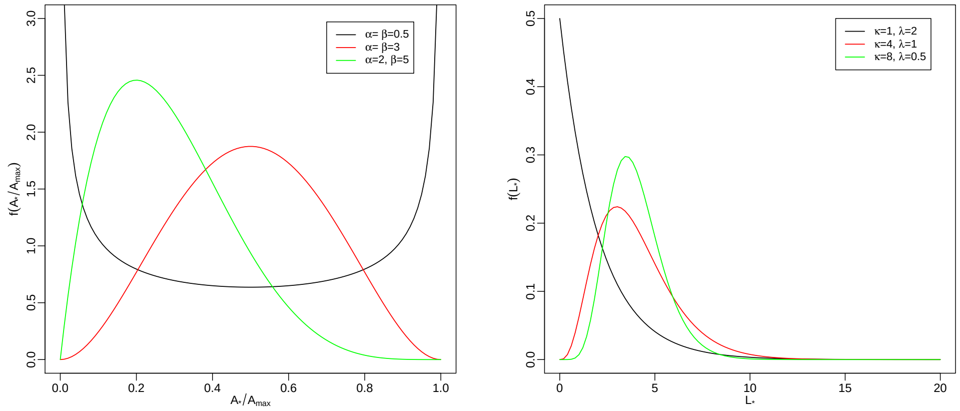


FIGURE 4.1: *Examples of the beta distribution (left), and examples of the gamma distribution (right).*

and when $\alpha, \beta > 1$

$$(iii) \quad \text{mode}\left(\frac{A_*}{A_{max}}\right) = \frac{\alpha - 1}{(\alpha + \beta - 2)}.$$

Figure 4.1(left) shows plots of the beta probability density function for various values of the parameters α and β. It is clear that for some combinations of α and β there are two modes and the distribution is U shaped. However, this is not sensible as such situations are unlikely where large area and small area are both likely. Therefore the bimodal model is not considered here, and at least one parameter of α or β should be greater than one.

In addition, let L_* be the total length of object boundaries. For more general situations, where not only the short length is more likely then, to model the length, a gamma distribution with parameters κ and λ can be used as a more appropriate distribution. Thus, the corresponding probability density function is

$$p(L_*) = \frac{1}{\Gamma(\kappa)} \lambda^{\kappa} L_*^{\kappa - 1} \exp\{-\lambda L_*\}, \qquad L_* \geq 0, \tag{4.2}$$

where $\Gamma(.)$ is the gamma function. If $L_* \sim \text{gamma}(\kappa, \lambda)$, then some properties of the gamma distribution are

$$(i) \quad E(L_*) = \frac{\kappa}{\lambda}, \qquad (ii) \quad \text{var}(L_*) = \frac{\kappa}{\lambda^2},$$

and when $\kappa \geq 1$,

$$\text{mode}(L_*) = \frac{(\kappa - 1)}{\lambda}.$$

Figure 4.1(right) shows plots of the gamma probability density function for various values of the parameters κ and λ. The gamma distribution with $\lambda = 1$ is an exponential distribution, and hence the standard model is a special case.

Assuming, for simplicity, that A_* and L_* are independent then the distributions in 4.1 and 4.2 can be combined to give the following joint prior distribution on the model parameters

$$\begin{aligned} p(\boldsymbol{\theta}) &= p(A_*) \times p(L_*) \\ &= \frac{\lambda^\kappa}{B(\alpha, \beta)\mathit{\Gamma}(\kappa)}(A_*/A_{max})^{\alpha-1}(1 - A_*/A_{max})^{\beta-1}L_*^{\kappa-1}\exp\{-\lambda L_*\}, \\ &\qquad 0 \leq A_* \leq A_{max} \qquad \text{and} \qquad L_* \geq 0. \qquad (4.3) \end{aligned}$$

Let $f(\boldsymbol{y}|\boldsymbol{\theta})$ be the conditional distribution of the observed image $\boldsymbol{y}$ given the parameter vector $\boldsymbol{\theta}$; let the noise be modelled by a Gaussian distribution with zero mean and variance σ^2. Consider the likelihood function in the Gaussian model, given by

$$f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma^2)^{N^2/2}}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i\in I_0}(y_i - \mu_0)^2 + \sum_{i\in I_*}(y_i - \mu_*)^2\right]\right\}, \qquad \sigma > 0. \qquad (4.4)$$

Using Bayes' theorem, the likelihood in 4.4 and prior distribution in 4.3 are then combined to give the following posterior distribution

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) &= \frac{C}{(\sigma^2)^{N^2/2}}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i\in I_0}(y_i - \mu_0)^2 + \sum_{i\in I_*}(y_i - \mu_*)^2\right]\right\} \\ &\qquad \cdot(A_*/A_{max})^{\alpha-1}(1 - A_*/A_{max})^{\beta-1}L_*^{\kappa-1}\exp\{-\lambda L_*\}, \qquad (4.5) \end{aligned}$$

where C represents the collected constant terms. Thus, the corresponding log posterior distribution is given by

$$\begin{aligned}
L(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}|\boldsymbol{y}) &= -\frac{N^2}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\Bigg[\sum_{i\in I_0}(y_i-\mu_0)^2 + \sum_{i\in I_*}(y_i-\mu_*)^2\Bigg] \\
&\quad +(\alpha-1)\log\frac{A_*}{A_{max}} + (\beta-1)\log\left(1-\frac{A_*}{A_{max}}\right) \\
&\quad -(\kappa-1)\log L_* - \lambda L_* + \log C. \qquad (4.6)
\end{aligned}$$

The MAP estimate is then defined as

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{y}) = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\boldsymbol{y}).$$

If the maximum is located at a turning points then $\hat{\boldsymbol{\theta}}_{MAP}$ can be found by differentiation. This gives the following equation system, which must be solved simultaneously, as explained in Chapter 3

$$\frac{\partial L}{\partial\varphi} = 0, \qquad \frac{\partial L}{\partial\mu_0} = 0, \qquad \frac{\partial L}{\partial\mu_*} = 0, \qquad \frac{\partial L}{\partial\sigma^2} = 0.$$

First, to estimate the level-set matrix $\boldsymbol{\varphi}$, the partial derivative of L with respect to $\boldsymbol{\varphi}$ is given by

$$\begin{aligned}
\frac{\partial L}{\partial\boldsymbol{\varphi}} &= \Big[-(\boldsymbol{y}-\mu_0)^2(\delta(\boldsymbol{\varphi})) + (\boldsymbol{y}-\mu_*)^2\delta(\boldsymbol{\varphi})\Big] + (\alpha-1)\frac{1}{A_{max}}.\frac{\delta(\boldsymbol{\varphi})}{A_*} \\
&\quad +(\beta-1)\delta(\boldsymbol{\varphi})\frac{A_{max}}{A_{max}-A_*} - (\kappa-1)\bigg(-\delta(\boldsymbol{\varphi})\mathrm{div}\Big(\frac{\nabla\boldsymbol{\varphi}}{|\nabla\boldsymbol{\varphi}|}\Big)\bigg)\frac{1}{L_*} \\
&\quad +\lambda\delta(\boldsymbol{\varphi})\mathrm{div}\Big(\frac{\nabla\boldsymbol{\varphi}}{|\nabla\boldsymbol{\varphi}|}\Big). \\
&= \delta(\boldsymbol{\varphi})\bigg[-(\boldsymbol{y}-\mu_0)^2 + (\boldsymbol{y}-\mu_*)^2 + (\alpha-1)\frac{1}{A_{max}A_*} + (\beta-1)\frac{A_{max}}{A_{max}-A_*} \\
&\quad +(\kappa-1)\mathrm{div}\Big(\frac{\nabla\boldsymbol{\varphi}}{|\nabla\boldsymbol{\varphi}|}\Big)\frac{1}{L_*} + \lambda\mathrm{div}\Big(\frac{\nabla\boldsymbol{\varphi}}{|\nabla\boldsymbol{\varphi}|}\Big)\bigg]. \qquad (4.7)
\end{aligned}$$

For fixed μ_0, μ_* and σ^2, and introducing an artificial time variable; then (as explained in Chapter 3) the discrete-time updating equation for the level-set matrix is

$$\boldsymbol{\varphi}^k = \boldsymbol{\varphi}^{k-1} + \Delta t \cdot \left.\frac{\partial \boldsymbol{\varphi}}{\partial t}\right|_{\boldsymbol{\varphi}^{k-1}}, \qquad \text{for} \quad k = 1, ... \tag{4.8}$$

To estimate the parameters μ_0, μ_*, and σ^2, the partial derivatives are

$$\frac{\partial L}{\partial \mu_0} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \mu_0} = \frac{1}{\sigma^2}\sum_{i\in I_0}(y_i - \mu_0),$$

$$\frac{\partial L}{\partial \mu_*} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \mu_*} = \frac{1}{\sigma^2}\sum_{i\in I_*}(y_i - \mu_*),$$

and

$$\frac{\partial L}{\partial \sigma^2} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \sigma^2} = -\frac{N^2}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\left[\sum_{i\in I_0}(y_i - \mu_0)^2 + \sum_{i\in I_*}(y_i - \mu_*)^2\right].$$

Setting each to zero leads to the following estimates

$$\begin{aligned} \hat{\mu}_0 &= \frac{1}{n_0}\sum_{i\in I_0} y_i, \qquad \hat{\mu}_* = \frac{1}{n_*}\sum_{i\in I_*} y_i, \qquad \text{and} \\ \hat{\sigma}^2 &= \frac{1}{N^2}\left[\sum_{i\in I_0}(y_i - \mu_0)^2 + \sum_{i\in I_*}(y_i - \mu_*)^2\right], \end{aligned} \tag{4.9}$$

where n_0 and n_* are the number of data points in regions I_0 and I_* respectively, and so $N^2 = n_0 + n_*$.

4.3 Choice of noise model

The aim here is to propose a robust alternative noise model to the Gaussian distribution. It is well known that a model with Gaussian likelihood is sensitive to outliers (McLachlan and Peel, 2004). Alternatively, a model with Student's t-distribution components can be used to fit a robust model. This is because the Student's t-distribution has heavier tails than the Gaussian distribution, hence it is not as sensitive to outliers. In particular, the following form will be used

$$g(x|\nu, \sigma) = \frac{\Gamma(\nu)}{\sqrt(\pi)\Gamma(\nu - 1/2)}\frac{\sigma^{\nu-1}}{(\sigma^2 + x^2/\nu)^{\nu}}, \qquad \nu > 1/2, \sigma > 0, \tag{4.10}$$

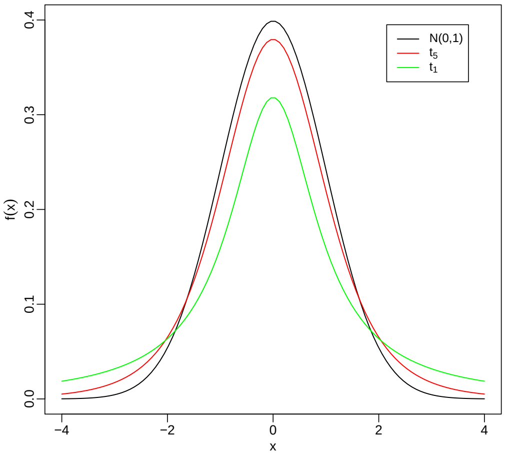


FIGURE 4.2: *Examples of the Student's t-distribution.*

where μ is the location parameter, σ is the scale parameter and ν is the degrees of freedom. Notice that the Student's t-distribution converges to the Gaussian distribution with mean μ and variance σ^2 when $\nu \to \infty$. Figure 4.2 shows three examples of standard Student's t-distributions with $\mu = 0$, $\sigma = 1$ and different values of the degrees of freedom ν. These examples are with $\nu = 1$, $\nu = 5$ and $\nu = \infty$. It is clear that for small values of ν, the distribution has heavier tails than that of a Gaussian distribution. Using the Student's t-distribution to replace the Gaussian distribution in Equation 4.4, the likelihood function is given by

$$\begin{aligned} f(\boldsymbol{y}|\boldsymbol{\theta}) &= \left(\frac{\Gamma(\nu)\sigma^{\nu-1}}{\sqrt(\pi)\Gamma(\nu-1/2)}\right)^{N^2} \prod_{i\in I_0}\left(\sigma^2+(y_i-\mu_0)^2/\nu\right)^{-\nu} \\ & \prod_{i\in I_*}\left(\sigma^2+(y_i-\mu_*)^2/\nu\right)^{-\nu}. \end{aligned} \tag{4.11}$$

The corresponding log-likelihood function is given by

$$\begin{aligned} \log f(\boldsymbol{y}|\boldsymbol{\theta}) &= \frac{N^2(\nu-1)}{2}\log\sigma^2 - \nu\left[\sum_{i\in I_0}\log\left(\sigma^2+(y_i-\mu_0)^2/\nu\right)\right. \\ & \left. +\sum_{i\in I_*}\log\left(\sigma^2+(y_i-\mu_*)^2/\nu\right)\right]. \end{aligned} \tag{4.12}$$

Now, assume that the prior distribution of the model parameters, $\boldsymbol{\theta}$, including A_* and L_* is given by

$$p(\boldsymbol{\theta}) = (\alpha\lambda)\exp\{-(\alpha A_* + \lambda L_*)\}, \qquad A_*, L_* \geq 0. \tag{4.13}$$

Then, using Bayes' theorem, the likelihood function and the prior distribution are combined to give the posterior distribution

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{f(\boldsymbol{y})}.$$

The corresponding log-posterior distribution is then given by

$$L(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}|\boldsymbol{y}) = \log f(\boldsymbol{y}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log f(\boldsymbol{y}). \tag{4.14}$$

Thus

$$\begin{aligned} L(\boldsymbol{\theta}) &= \frac{N^2(\nu-1)}{2}\log\sigma^2 - \nu\left[\sum_{i\in I_0}\log\left(\sigma^2 + (y_i-\mu_0)^2/\nu\right)\right. \\ &\quad \left. + \sum_{i\in I_*}\log\left(\sigma^2 + (y_i-\mu_*)^2/\nu\right)\right] - (\alpha A_* + \lambda L_*) + C, \end{aligned} \tag{4.15}$$

where C is a constant, and again the MAP estimate is given by

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}).$$

Similarly to the previous case, to estimate $\boldsymbol{\varphi}$, the partial derivative of $\log p(\boldsymbol{\theta} \mid \boldsymbol{y})$ with respect to $\boldsymbol{\varphi}$ is given by

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{\varphi}} &= \delta(\boldsymbol{\varphi})\left[-\log\left(\sigma^2 + (y_i-\mu_0)^2/\nu\right) + \log\left(\sigma^2 + (y_i-\mu_*)^2/\nu\right)\right. \\ &\quad \left. + \alpha - \lambda \text{div}\left(\frac{\nabla\boldsymbol{\varphi}}{|\nabla\boldsymbol{\varphi}|}\right)\right]. \end{aligned} \tag{4.16}$$

For fixed μ_0, μ_* and σ^2, and introducing an artificial time variable; then (as explained in Chapter 3) the discrete-time updating equation for the level-set matrix is

$$\boldsymbol{\varphi}^k = \boldsymbol{\varphi}^{k-1} + \Delta t \cdot \left.\frac{\partial\boldsymbol{\varphi}}{\partial t}\right|_{\boldsymbol{\varphi}^{k-1}}, \qquad \text{for} \quad k = 1, ... \tag{4.17}$$

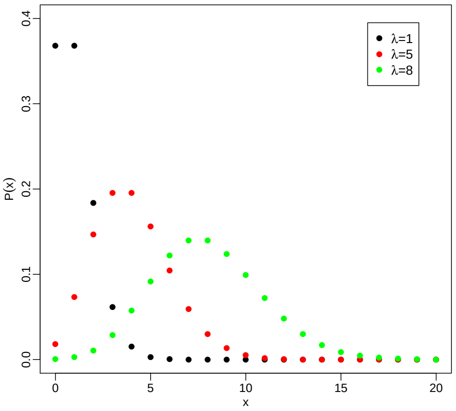


FIGURE 4.3: *Examples of the Poisson distribution.*

The rest of the parameters μ_0, μ_* and σ^2 are estimated through function `optim` implemented in R. This is because the log-posterior in Equation 4.15 is too complicated and the partial derivatives with respect to those parameters cannot be obtained analytically.

Another possible data modelling situation is when the data are discrete, examples of this type of data can be found in medical imaging (see Weir and Green (1994) and the references there). One possible choice is that the errors are from a Poisson distribution with a single parameter λ, which has the following probability mass function

$$g(x|\lambda) = \frac{\lambda^x \exp\{-\lambda\}}{x!}, \qquad x \in 0, 1, \dots \qquad \lambda > 0. \tag{4.18}$$

Note that if $\lambda \to \infty$, then the Poisson distribution converges to the Gaussian distribution with the mean λ and variance λ. Figure 4.3 shows three examples of Poisson distributions with different values of the parameter λ. In particular, the three density functions shown in the figure are with $\lambda = 1$, $\lambda = 4$ and $\lambda = 8$. It is clear that, for small values of λ, the most likely value is zero. As λ increases, zero becomes less likely, and the distribution tends to be more symmetric. The likelihood function, based on 4.18, is given by

$$f(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{i \in I_0} \frac{\mu_0^{y_i} \exp\{-\mu_0\}}{y_i!} \prod_{i \in I_*} \frac{\mu_*^{y_i} \exp\{-\mu_*\}}{y_i!}. \tag{4.19}$$

The corresponding log-likelihood function is given by

$$\log f(\boldsymbol{y}|\boldsymbol{\theta}) \quad = \quad \log \mu_0 \sum_{i \in I_0} y_i - n_0\mu_0 + \log \mu_* \sum_{i \in I_*} y_i - n_*\mu_*. \tag{4.20}$$

Assume that the prior distribution of the model parameters, $\boldsymbol{\theta}$, including A_* and L_* is given by

$$p(\boldsymbol{\theta}) = (\alpha\lambda)\exp\{-(\alpha A_* + \lambda L_*)\}, \qquad A_*, L_* \geq 0. \tag{4.21}$$

Using Bayes' theorem, the likelihood function and the prior distribution are then combined to give the posterior distribution

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{f(\boldsymbol{y})}.$$

The corresponding log-posterior distribution is then given by

$$L(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}|\boldsymbol{y}) = \log f(\boldsymbol{y}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log f(\boldsymbol{y}). \tag{4.22}$$

Thus

$$L(\boldsymbol{\theta}) \quad = \quad \left(\log \mu_0 \sum_{i \in I_0} y_i - n_0\mu_0 + \log \mu_* \sum_{i \in I_*} y_i - n_*\mu_* \right) - (\alpha A_* + \lambda L_*) + C, \tag{4.23}$$

where C is a constant. The MAP estimate is given by

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}).$$

As in the previous case, $\boldsymbol{\varphi}$ can be estimated by fixing the parameters μ_0, μ_* and σ^2; then, the partial derivative of $\log p(\boldsymbol{\theta} \mid \boldsymbol{y})$ with respect to φ is given by

$$\frac{\partial L}{\partial \boldsymbol{\varphi}} \quad = \quad \delta(\boldsymbol{\varphi})\left[-\log \mu_0 \sum_{i \in I_0} y_i + \log \mu_* \sum_{i \in I_*} y_i + \alpha - \lambda \text{div}\left(\frac{\nabla \boldsymbol{\varphi}}{| \nabla \boldsymbol{\varphi}|} \right) \right]. \tag{4.24}$$

For fixed μ_0 and μ_*, and introducing an artificial time variable; then (as explained in Chapter 3) the discrete-time updating equation for the level-set matrix is

$$\boldsymbol{\varphi}^k = \boldsymbol{\varphi}^{k-1} + \Delta t \cdot \left.\frac{\partial \boldsymbol{\varphi}}{\partial t}\right|_{\boldsymbol{\varphi}^{k-1}}, \qquad \text{for} \quad k = 1, ... \tag{4.25}$$

Again, to find a solution for μ_0, and μ_*, based on the current level-set $\boldsymbol{\varphi}^k$ the partial derivatives are

$$\frac{\partial L}{\partial \mu_0} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \mu_0} = \sum_{i \in I_0} (y_i - \mu_0),$$

and

$$\frac{\partial L}{\partial \mu_*} = \frac{\partial(\log p(\boldsymbol{\theta}|\boldsymbol{y}))}{\partial \mu_*} = \sum_{i \in I_*} (y_i - \mu_*).$$

Setting each to zero leads to the following estimates

$$\hat{\mu}_0 = \frac{1}{n_0} \sum_{i \in I_0} y_i, \quad \text{and} \quad \hat{\mu}_* = \frac{1}{n_*} \sum_{i \in I_*} y_i.$$

where n_0 and n_* are the number of data points in regions I_0 and I_* respectively, and so $N^2 = n_0 + n_*$.

4.4 Simulation studies

The aim here is to investigate the robustness of the models in cases where the data are generated from other models, or when the data contain outliers. Also, the sensitivity to the choice of prior parameters in the beta distribution is studied. The results are compared using various posterior measures, such as posterior area, and length. Figure 4.4 shows the posterior mode for area over a range of area parameters α and β. It is clear that the most likely values for beta prior distribution are when α and β are relatively equal, and hence when the distribution is symmetric.

The proposed models use Student's t-distributions and Poisson distributions as the alternative likelihood functions to the Gaussian distribution. These are then applied to simulated datasets, investigating the suitability for image segmentation problems. In particular, the new models are used to analyse the cut-circle image data and compared with the fit of the Gaussian model.

A synthetic image that contains a cut-circle object was considered; 8 datasets were simulated from the Poisson distribution with $\mu_1 = 2$ and μ_0 a variable multiple of μ_1 (i.e., $\mu_0 = \{k\mu_1 : k = 3, \ldots, 10\}$). Figure 4.5 shows two simulated

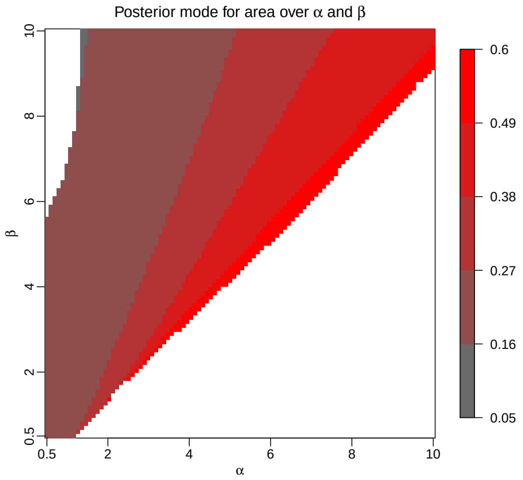


FIGURE 4.4: *The posterior density for the object area over a range of area parameters α and β.*

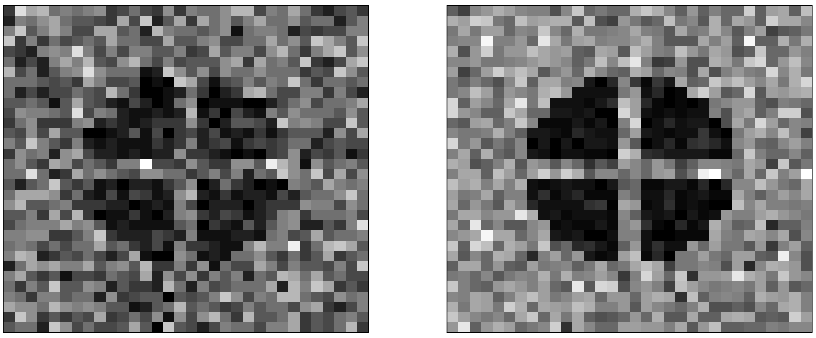


FIGURE 4.5: *Example of datasets with: $\mu_1 = 2$ and $\mu_0 = 3\mu_1$ (left) and $\mu_0 = 7\mu_1$ (right).*

datasets for different values of μ_0, note that when the value of k is less than 3 then the segmentation becomes too difficult. This is because the contrast is very small. Similarly, when k is bigger than 10 the problem becomes very easy.

The Poisson and the Gaussian models were fitted. The CCR and the SE for the area of the object and the length of the boundary were calculated. The results from both fitted models are shown in Figure 4.6. In each panel, the black dotted line represents the results from the Gaussian fitted model, and the red dotted line represents the results from the Poisson fitted model. Panel (a) in the figure shows that for small values of k, the CCR for the Poisson fitted model is higher than the Gaussian fitted model. As k increases, the CCR for the two

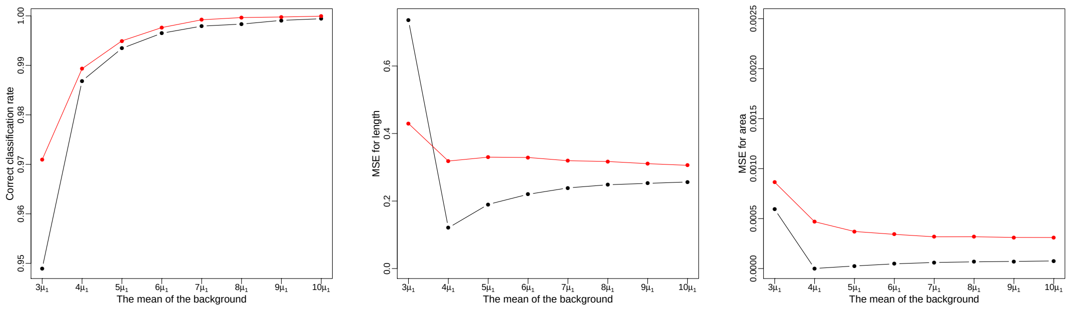


FIGURE 4.6: *The correct classification rate (left), the SE for the area of the object (centre) and the SE for the length of the boundaries (right). The results from the Gaussian fitted model are shown with the black line, whilst the Poisson fitted model is shown with the red line.*

fitted models become close. At $k = 10$ they become equal. The SE results for the area of the object and the length of the boundaries are shown in (b) and (c), respectively. It is clear that all SE values for the area are very small, showing little difference between the two fitted models. The SE for the length shows that for small values of k, the Poisson fitted model is better with lower SE values. At $k = 4$ there is a huge jump in SE for the Gaussian fitted model. As k increases, the SE values from both fitted models become equal. Therefore, the results based on the CCR show a more robust fitted model by using the Poisson distribution to model the count data.

The robustness of the Student's t likelihood model is now investigated. The data used here were generated from the Gaussian model, with the background intensity $\mu_0 = 10$, the object intensity $\mu_* = 5$ and common noise variance $\sigma^2 = 1$. In order to evaluate the robustness of the image segmentation, about 10% of the image pixels were drawn from a Gaussian model with higher variance. Then the Gaussian model and the Student's t model are fitted to the experimental data. In particular, the data were analysed first by fitting the Gaussian model, then the leave-one-out approach was applied to identify the influential observations. All those influential observations were weighted as explained in Chapter 3, and then the Gaussian model with weighted likelihood was fitted. Finally, the data was analysed by fitting the Student's t model.

Figure 4.7 shows segmentation results for the cut-circle image with 10% of the

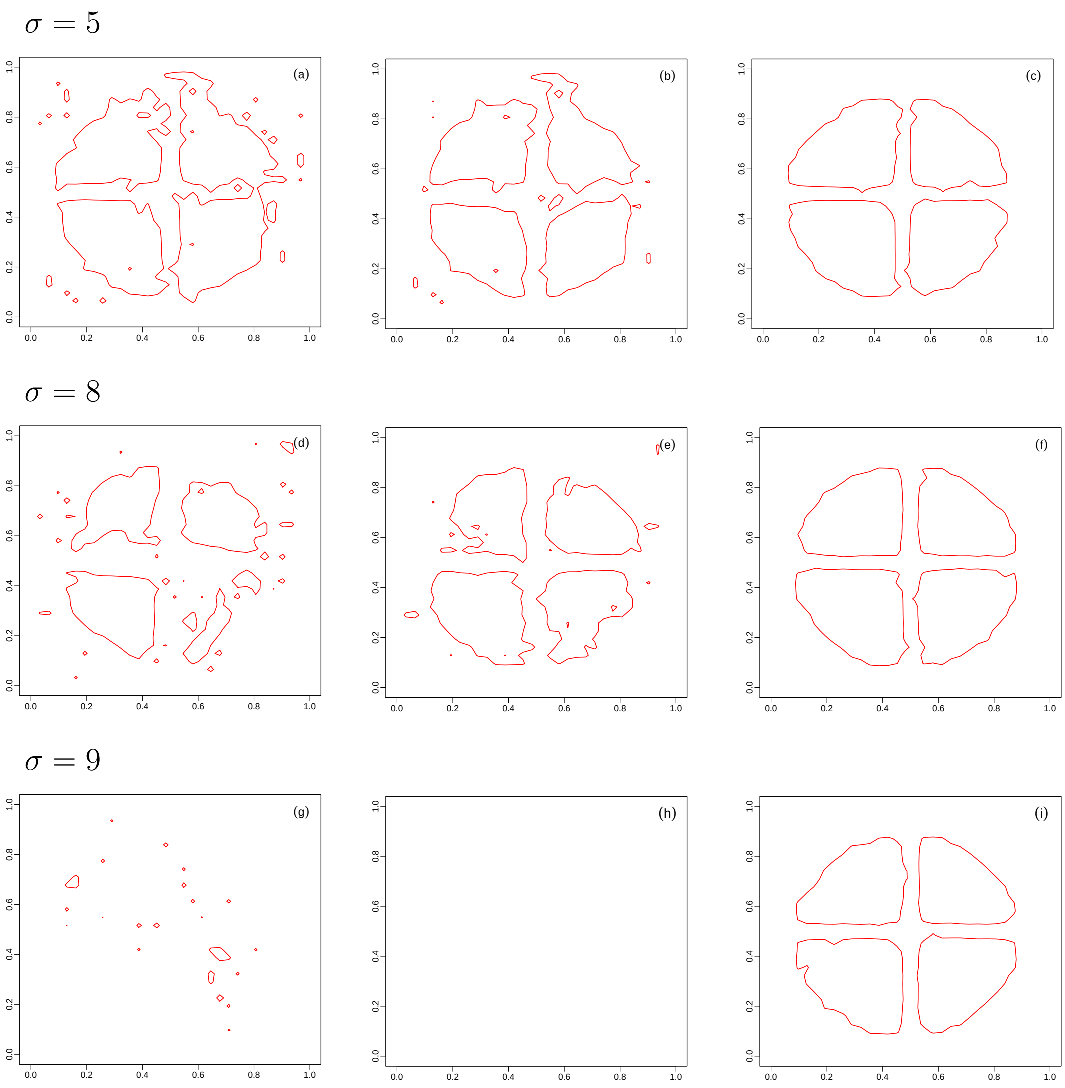


FIGURE 4.7: *Segmentation results of the cut-circle image with* 10% *of the image generated from the Gaussian distribution with different standard deviation values. The final zero level-set using the Gaussian model (first column), the final zero level-set using the weighted Gaussian model (middle column), and the final segmented image using the Student's t model (last column) are shown.*

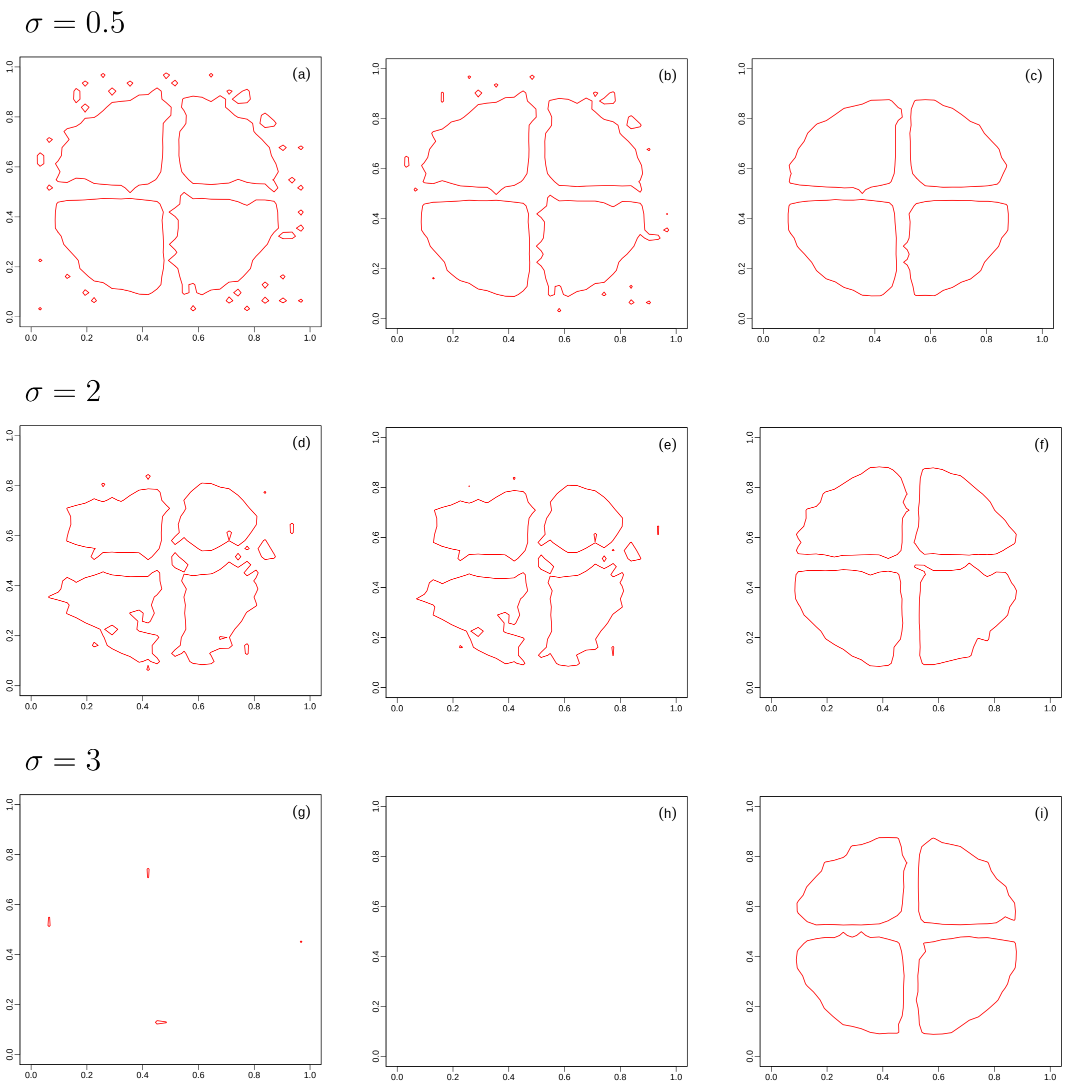
image from the Gaussian distribution with standard deviation $\sigma = 5$, $\sigma = 8$ and $\sigma = 9$. The figure shows the final zero level-set from the Gaussian fitted model (first column), from the weighted Gaussian fitted model (second column) and from the Student's t fitted model (third column). It is clear from the figure that when the observations are from the Gaussian distribution with $\sigma = 5$, the cut-circle object in (a) was identified but with lots of fragments. After fitting the weighting Gaussian model, (b), the segmentation results seem fine but with some fragments still remaining. However, with the Student's t model the results are substantially improved in (c) and the cut-circle object was segmented

	CCR	SE for the area	SE for the length
$\sigma = 5$			
Gaussian model	0.908	0.0002	7.97
Weighted Gaussian model	0.898	0.0040	0.10
Student's t model	0.978	0.0002	0.04
$\sigma = 8$			
Gaussian model	0.803	0.0400	0.98
Weighted Gaussian model	0.897	0.0400	0.37
Student's t model	0.987	0.0001	0.06
$\sigma = 9$			
Gaussian model	0.612	0.1800	17.75
Weighted Gaussian model	-	-	-
Student's t model	0.983	0.0006	00.08

TABLE 4.1: *The CCR and the SE for the area of the object and the length of its boundaries from fitting the Gaussian model using unweighted and weighted data and the Student's t model.*

very well.

Then, with 10% of the image from the Gaussian distribution with $\sigma = 8$, the segmentation results obtained by fitting the Gaussian model, (d), and the weighted Gaussian model, (e), are very poor. On the other hand, the segmentation result obtained by fitting the Student's t model in (f) shows that the cut-circle object was identified with the four parts perfectly. Finally, when the observations were generated from the Student's t distribution with $\sigma = 9$, only the Student's t model successfully identified the cut-circle object, (i), where the evolving curve in (g) collapsed when using the Gaussian model. Hence, the influential observations cannot be identified and the model with weighted likelihood cannot be fitted as it is impossible to estimate the weights. The CCR, as well as the SE for the object area and the length of the boundaries are shown in Table 4.1. The numerical results support that fitting the Student's t distribution model gives better results with a higher CCR compared to the Gaussian and the weighted Gaussian models.


$\sigma = 0.5$

$\sigma = 2$

$\sigma = 3$

FIGURE 4.8: *Segmentation results of the cut-circle image with* 10% *of the image generated from the Student's t-distribution with different scale parameter values. The final zero level-set using the Gaussian model (first column), the final zero level-set using the weighted Gaussian model (middle column), and the final segmented image using the Student's t model (last column) are shown.*

Figure 4.8 shows segmentation results of the cut-circle image with 10% of the image from the Student's t-distribution with the degrees of freedom $\nu = 1$, a location parameter $\mu = 0$, and scale parameter values $\sigma = 0.5$, $\sigma = 2$ and $\sigma = 3$. The figure shows the final zero level-set from the Gaussian fitted model (first column), from the weighted Gaussian fitted model (second column) and from the Student's t fitted model (third column). It is clear that with $\sigma = 0.5$, the cut-circle object is identified using the standard Gaussian model, (a), but with lots of fragments. After fitting the weighted Gaussian model, the segmentation

	CCR	SE for the area	SE for the length
$\sigma = 0.5$			
Gaussian model	0.930	0.0004	7.88
Weighted Gaussian model	0.947	0.0001	1.81
Student's t model	0.986	0.0000	0.07
$\sigma = 2$			
Gaussian model	0.839	0.0200	1.1000
Weighted Gaussian model	0.847	0.0200	1.1200
Student's t model	0.973	0.0003	0.0005
$\sigma = 3$			
Gaussian model	0.595	0.1800	25.66
Weighted Gaussian model	-	-	-
Student's t model	0.983	0.0001	00.10

TABLE 4.2: *The CCR and the SE for the area of the object and the length of its boundaries from fitting the Gaussian model, the weighted Gaussian model, and the Student's t model.*

results seem fine but with some fragment still remaining in (b). However when fitting the Student's t model the results are improved as in (c).

Then, with 10% of the image from the Student's t-distribution with $\sigma = 2$, the segmentation results obtained by fitting the Gaussian model, (d), and the weighted Gaussian model, (e), are very poor. On the other hand, the segmentation result obtained by fitting the Student's t model in (f) shows that the cut-circle object is identified with the four parts perfectly. Finally, when the observations are generated from the Student's t-distribution with $\sigma = 3$, only the Student's t model successfully identifies the cut-circle object as in (i), whereas the curve collapses, (g), when using the Gaussian model. Thus, the influential observations cannot be identified and the model with weighted likelihood cannot be fitted as it is impossible to estimate the weights. Summary numerical results are shown in Table 4.2. Clearly, these results show that there is a significant improvement in segmentation when using the Student's t model.

4.5 Conclusions

The proposed models in this chapter use a Bayesian level-set framework, which is described in Chapter 3, based on the standard Gaussian likelihood and with exponential prior distributions on object area and boundary length. An extension was proposed using a weight function, this is to account for unreliable data. The corresponding model was applied to simulated data showing its suitability for image segmentation problems. Also as a robust alternative to the Gaussian distribution, the Student's t-distribution was proposed as the data model. In addition, the beta distribution was used as a more suitable distribution for modelling the area of the objects, and the gamma distribution was used as a more flexible generalization for modelling the length of the boundaries, and the model has been verified on simulated data. The results showed that there is a significant improvement in goodness of fit when using the Student's t model over that of the Gaussian model and the weighted Gaussian model.

Chapter 5

Markov chain Monte Carlo techniques for image segmentation

5.1 Introduction

In image segmentation the aim is to partition an observed image into several homogeneous regions, which are often known as the background and the objects. This can be done by fitting appropriate models, and in particular in the Bayesian approach by defining a likelihood and a prior distribution, which are combined using Bayes' theorem to give a posterior distribution which is then used as the basis for parameter estimation. For background on Bayesian modelling see for example Besag et al. (1995) and Gelman et al. (2014). As an alternative to using an analytic estimation approach in complicated Bayesian modelling problems, numerical algorithms based on Markov chain Monte Carlo (MCMC) techniques are often used. In addition to parameter estimation and image segmentation results, these algorithms allow for more general forms of modelling such as different types of data models and prior distributions, and output summary. In addition, the MCMC method allows a full investigation into estimation reliability and uncertainty (Aykroyd, 2015).

In this chapter, the LSM and MCMC techniques are combined to produce a new model fitting approach to object identification in image analysis. The key unknown parameter is the interface between the background and the objects.

The object boundary, in the level-set formulation, is defined as the zero level-set curve of the level-set matrix. The goal is to estimate this matrix based on Bayesian modelling and using the MCMC estimation algorithm. Combining the level-set method and MCMC has been studied by others. In particular, Xie et al. (2011) present a new method for history matching and uncertainty quantification for channelized reservoir models using the LSM and MCMC; however, the method's concepts are not simple and so it is not easy to implement. In the paper by Iglesias et al. (2015), a level-set based approach to Bayesian geometric inverse problems is introduced leading to a well-posed inverse problem; however, the numerical results presented show that initialization of the MCMC method for the level-set function can have a significant impact on the performance of the inversion technique. The method is also sensitive to the choice of prior parameters, which suggests studying hierarchical Bayesian modelling. The recent paper of Dunlop et al. (2017) studies automatic selection techniques for the prior parameters. In particular, the paper shows that Bayesian level-set inversion is improved significantly by a hierarchical Bayesian method in which the prior parameters are estimated.

Comparing to the previous works which combine the level-set method and MCMC methods in image segmentation, this is the first work which proposes a flexible way to simulate boundaries from the posterior distribution and which allows a fuller investigation than existing methods. In addition, the proposed algorithm is easy to describe and simple to apply. In contrast to other works, a wider range of the output summaries are shown. This includes for example, the credible intervals for the boundary regions, and the posterior probability for pixels being from the object regions. Furthermore, the proposed narrow-band updates presented here provide a quicker approach without loss of estimation accuracy and algorithm stability. The proposed algorithm has been applied to a variety of challenging examples in image segmentation, whereas other works have applied their methods to segment only simple shaped objects.

The likelihood of the data given the unknown parameters and prior distribution of the parameters are combined, using Bayes' theorem, to give the posterior distribution. The proposed method aims to segment a given image into different homogeneous regions by estimating this posterior distribution. The estimation procedure and the image segmentation are based on a level-set approach, which assumes the implicit representation of the object outlined as the zero level-set contour of a higher dimensional function. To this end, the Metropolis–Hastings (M–H) algorithm is used to estimate the level-set matrix. The proposed model will be in a Bayesian framework, combining noisy measurements and prior information about the level-set matrix. As well as estimates of the level-set matrix, the MCMC approach allows the estimation of the zero level-set curve, the mean intensity and the variance of both the background and the objects. Also, the flexibility of the MCMC method allows a more general investigation of the posterior distribution. In real applications of image segmentation, it is often the case that the number of objects is unknown; estimation of the object number and other model parameters is therefore also an aim.

The next section gives basic definitions, then Section 5.3 descibes the proposed Bayesian modelling. The general background of MCMC methods and the M–H algorithm in particular are given in Section 5.4. The use of the M–H algorithm to estimate the model parameters is described in Section 5.5. In Section 5.6 a detailed description of some computational issues is given. The proposed algorithm is tested on a wide variety of grey-level simulated images and the numerical results are shown in Section 5.7. Finally, the proposed method is applied to a real dataset in Section 5.8.

5.2 Definitions

Recall the definitions in Chapter 3 that a $2D$ space, $\boldsymbol{S} \in \mathbb{R}^2$, is divided into $N \times N$ pixels, and the unknown intensity is defined as $x(\boldsymbol{s})$ where $s \in \boldsymbol{S}$. Also, a finite set of measurements $\boldsymbol{y} = \{y_i : i = 1, \dots, N^2\}$ are collected at locations $\boldsymbol{s} = \{s_i : i = 1, \dots, N^2\}$ within the domain $\boldsymbol{S}$. Often these locations will form a finite regular square lattice, and the data can be displayed as a grey-level image. Furthermore, the intensity function can be discretized onto the same square lattice to give unknown variables $\boldsymbol{x} = \{x_i : i = 1, \dots, N^2\}$ representing the value x_i at the measured points, that is $x_i = x(\boldsymbol{s}_i)$ for $i = 1, \dots, N^2$, and can also be displayed as a grey-level image.

Now assume that the domain is divided into non-overlapping regions, $\boldsymbol{R} = \{R_k : k = 0, \dots, m-1\}$, with corresponding characteristic intensities $\boldsymbol{\mu} = \{\mu_k :$

$k = 0, \ldots, m-1\}$. Hence $R_k \cap R_l = \emptyset$, for $k \neq l$, and $R_0 \cup \ldots \cup R_{m-1} = \boldsymbol{S}$. These regions are labelled background, R_0, and objects, $R_* = R_0^c$. R_1 is the first object, R_2 is the second object, and so on.

The measurements can be defined in terms of a deterministic component, which only depends on the region, and a stochastic component, i.e.

$$y_i = x_i + \epsilon_i \qquad i = 1, ..., N^2, \tag{5.1}$$

where

$$x_i = \begin{cases} \mu_0 & \text{if} \quad s_i \in R_0, \\ \mu_* & \text{if} \quad s_i \in R_*, \end{cases} \tag{5.2}$$

and the errors ϵ_i are independently distributed from $N(0, \sigma^2)$ for $i = 1, \ldots, N^2$.

5.3 Bayesian modelling

In the classical inference approach, parameters are considered as fixed but unknown values; however, in the Bayesian analysis, these parameters are considered as random variables described by probability distributions (Davison, 2003). The posterior probability of these parameters given the data and prior information is proportional to the product of likelihood and prior probability. In practice, evaluating the posterior probability is almost impossible. This is because the unknown parameters are usually of high dimension and so the posterior distribution is very complex making the integration required for correct normalization impractical. However, the M-H algorithm can be used to produce approximate samples, from the posterior distribution from which the features of the posterior distribution are explored and the parameters are estimated.

The aim here is to segment a domain of interest into homogeneous regions, which are the background and the objects. Let γ be the outline of the objects; that is, the interface between the background and the objects. In the level-set formulation, the curve γ can be defined as a zero level of the level-set matrix $\boldsymbol{\varphi}$. Thus, estimating the level-set function will provide the required segmentation and so the model parameter set is $\boldsymbol{\theta} = \{\boldsymbol{\varphi}, \mu_0, \mu_*, \sigma^2\}$.

The image data can be modelled as follows. Consider a conditional distribution $f(\boldsymbol{y}|\boldsymbol{\theta})$ of the observed data $\boldsymbol{y} = \{y_i : i = 1, \ldots, N^2\}$ given the parameters set $\boldsymbol{\theta} = \{\boldsymbol{\varphi}\}$. The likelihood is given by

$$f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{\left(2\pi\sigma^2\right)^{N^2/2}} \exp\left\{ -\frac{1}{2\sigma^2}\left[\sum_{i\in I_0}(y_i - \mu_0)^2 + \sum_{i\in I_*}(y_i - \mu_*)^2\right]\right\}, \tag{5.3}$$

where $I_0 = \{i : \varphi_i < 0\}$ and $I_* = \{i : \varphi_i \geq 0\}$.

The parameters are modelled by a prior density, which quantifies any prior information about them. This prior density should give high probabilities to parameter values which agree with the prior information, and low probabilities to unacceptable parameter values. The general beliefs about the level-set matrix $\boldsymbol{\varphi}$ are that there is some spatial prior information, reflecting the idea that the neighbouring components in the level-set matrix are likely to have similar values, which suggests smoothness. Also there is an important numerical issue, which is that the component values of $\boldsymbol{\varphi}$ need to be controlled from drifting into extremely positive or negative values through the updating scheme. In other words, there is a need for a function to shrink the values. Thus, consider the case where each parameter value φ_i follows a Gaussian distribution with mean zero and standard deviation τ (which controls the amount of shrinkage). It is assumed that there is no prior information about the other model parameters leadings to improper uniform prior distribution. All this information leads to a prior density function with two components

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\varphi}) \propto \exp\left\{ -\frac{1}{2\tau^2}\sum_{i=1}^{N^2}\varphi_i^2 - \frac{1}{2\beta^2}\sum_{i\sim j}(\varphi_i - \varphi_j)^2\right\}, \tag{5.4}$$

where $i \sim j$ denotes the sum over pairs of neighbours in a four nearest neighbour system on the square lattice, and τ controls the amount of shrinkage. Here first-order smoothing is considered with smoothing parameter β, which defines the level of variability between neighbouring component values. Note that as $\beta \to 0$ all components tend to be similar and the image becomes constant, and as $\beta \to \infty$ the component values are spatially independent. The function in Equation 5.4 is similar to that of the fused lasso in regression (Friedman et al., 2001).

The product of the likelihood and the prior distributions using Bayes' theorem defines the posterior distribution as

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{f(\boldsymbol{y})},$$

where $f(\boldsymbol{y})$ is the normalizing constant. Inference about the model parameter is based on this posterior distribution. However, when the unknown parameter vector is high dimensional, the posterior distribution can become complicated and so it is impractical to do the inference analytically, but, numerical methods can be used for such estimation problems. One of the most widely used techniques is the MCMC method. Here the proposed algorithm is based on the MCMC method, and aims to estimate the level-set matrix $\boldsymbol{\varphi}$, μ_0, μ_* and σ^2 with the parameters τ and β assumed to be known.

5.4 Markov chain Monte Carlo methods

Markov chain Monte Carlo (MCMC) methods were invented for physics applications, specifically to compute complex integrals. These integrals can be formed as expectations, which are estimated using generated samples from a target distribution. MCMC methods are used in statistics as a simple and flexible technique allowing more general and complicated models to be considered. The key aim of MCMC methods is to explore the properties of a given distribution efficiently. An early review of MCMC methods is given in Besag et al. (1995), and many practical applications can be found in Gilks et al. (1995), Liu (2008) and Voss (2013). The recent paper by Robert and Casella (2011) gives a comprehensive review of the development of MCMC methods. Furthermore, a review of statistical approaches in image analysis applications can be found in Aykroyd (2015).

In the Bayesian context, consider first a distribution of the observed data $\boldsymbol{y}$ given the parameter vector $\boldsymbol{\theta}$; this is the likelihood function denoted by $f(\boldsymbol{y}|\boldsymbol{\theta})$. Second, a distribution of $\boldsymbol{\theta}$; that is, the prior distribution $p(\boldsymbol{\theta})$. These distributions can be combined to give the distribution of the updated parameter $\boldsymbol{\theta}$ given the data $\boldsymbol{y}$. This posterior distribution, denoted by $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, is expressed as follows:

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int f(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}.$$

In complicated modelling, it is difficult to find the integral $\int f(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ analytically. However, MCMC methods offer an alternative way to simulate samples from the posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ and obtain estimates of the quantities of interest. In MCMC terminology, the posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ is called the target distribution. Essentially, the MCMC method is a generalisation of the Monte Carlo method where the independent samples are replaced by correlated samples from the posterior distribution. In particular, MCMC samples are generated sequentially; the next sampled value depends only on the sampled value most recently generated.

One technique, which is a special case of MCMC methods is, the Metropolis-Hastings (M-H) algorithm (Metropolis et al., 1953; Hastings, 1970), which is now one of the most popular sampling algorithms. The aim of this algorithm is to simulate a Markov chain which has the required posterior distribution as its equilibrium distribution. A good review of the M-H algorithm is given by Chib and Greenberg (1995). A summary of the standard algorithm can be found in Figure 5.1.

The MCMC approach gives a framework which can be used in many estimation problems. The algorithm is used to produce a correlated sample from some target statistical distribution (usually the posterior distribution in a Bayesian analysis). If the transitions in the Markov chain are designed well, then the simulated sample will have the same statistical properties as a sample obtained directly from the target distribution. However, during any run there is a burn-in period after which the chain settles. Thus, to reduce the effect of bias, the generated sample from the beginning are ignored and only the remaining sample is included in the parameter estimation. For example, consider the sample $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$ from N iterations. After $M \ll N$ iterations, the chain is said to have converged and it is approximately generated from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$. Thus, with regards to the aim of estimating the parameters, only the observations $\{\boldsymbol{\theta}_{M+1}, \boldsymbol{\theta}_{M+2}, \ldots, \boldsymbol{\theta}_N\}$ are used.

Initialise with an arbitrary value $\boldsymbol{\theta}^0$.

Repeat the following steps for $k = 1, \ldots, K$

1. Generate a proposal $\boldsymbol{\theta}^*$ from the proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{k-1})$.
2. Evaluate the acceptance probability $\alpha(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{k-1})$ in Equation 5.5.
3. Generate u from a uniform distribution $U(0, 1)$. If $u \leq \alpha$, the proposed $\boldsymbol{\theta}^*$ is accepted, and $\boldsymbol{\theta}^k = \boldsymbol{\theta}^*$. Otherwise, $\boldsymbol{\theta}^*$ is rejected, and $\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1}$.

FIGURE 5.1: *The standard M-H algorithm.*

One of the main issues to be considered in MCMC methods is the number of steps required for the chain to converge. In statistics literature, there are several suggestions for how long the MCMC algorithm should be run until the chain converges. Gilks et al. (1995) provided one such idea that involves running the chain with different starting points and comparing the results. Another idea is to calculate autocorrelations; the chain reaches equilibrium when covariances begin to decay (Geyer, 1992).

The M-H algorithm can be used to generate samples from the target distribution, $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, as follows. At each iteration k, a candidate value, or proposal, $\boldsymbol{\theta}^*$, is drawn from a proposal distribution $q(\boldsymbol{\theta}^{k-1}, \boldsymbol{\theta}^*)$, where $\boldsymbol{\theta}^{k-1}$ is the current state. The proposal is accepted as the next state of the chain with the following acceptance probability:

$$\alpha(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = \min\left[1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{y})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|\boldsymbol{y})q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}\right]. \tag{5.5}$$

Otherwise, the proposal $\boldsymbol{\theta}^*$ is rejected, the chain does not move, and the current value $\boldsymbol{\theta}$ is taken as the value of the next state. The most usual choice of the proposed distribution is the normal distribution, with the mean as the current state value and an appropriate variance value. The normal distribution is symmetric, and so the expression for the acceptance probability α becomes simpler, as $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ and $q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ cancel out.

The choice of the variance of the proposal distribution is important to ensure the efficiency of the algorithm. If the proposal variance is small, then the acceptance rate is large and almost all proposals are accepted; thus the chain moves in small steps and will take a long time to converge. Conversely, if the proposal variance is large, then the acceptance rate is small, and many proposals are rejected, and the chain does not move often. Hence the proposal variance can be chosen by monitoring the acceptance rate. In Roberts et al. (1997), the optimal rate was determined to be 23.4% for a wide range of high dimensional examples, whereas in low dimensional problems the acceptance rate can be around 44% (Gelman et al., 1996).

5.5 Parameter estimation via the M-H algorithm

The properties of the posterior distribution of the proposed model can be studied using the M-H algorithm to generate samples of the level-set matrix through an iterative procedure. In particular, the M-H algorithm simulates a Markov chain for the level-set matrix $\boldsymbol{\varphi}$. As the level-set matrix $\boldsymbol{\varphi}$ consists of all components φ_i, where $i = 1, \ldots, N^2$, it might be computationally efficient to propose and update these parameters one by one (Aykroyd, 2015); this is known as a single-component M-H algorithm (Gilks et al., 1995). The M-H algorithm proposed here is based on a random walk perturbation. The proposal distribution considered is the normal distribution centred on the current state (the current value of the parameter), with the proposal variance κ^2 chosen adaptively through the burn-in period to give a reasonable acceptance rate. The single component M-H algorithm used here can be described as follows: at iteration $k-1$ let the current parameter be $\boldsymbol{\varphi}^{k-1} = \{\varphi_1^{k-1}, \ldots, \varphi_i^{k-1}, \ldots, \varphi_{N^2}^{k-1}\}$, then all components φ_i are considered separately in the updating algorithm. For each i, a new value φ_i^* is proposed as a perturbation of the previous value φ_i^{k-1}, then $\varphi_i^* = \varphi_i^{k-1} + \epsilon$, where $\epsilon \sim N(0, \kappa^2)$. The proposed value φ_i^* is accepted with probability

$$\alpha(\varphi_i^* | \varphi_i^{k-1}) = \min\left[1, \frac{\pi(\varphi_1^k, \varphi_2^k, \ldots, \varphi_{i-1}^k, \varphi_i^*, \varphi_{i+1}^{k-1}, \ldots, \varphi_{N^2-1}^{k-1}, \varphi_{N^2}^{k-1} | \boldsymbol{y})}{\pi(\varphi_1^k, \varphi_2^k, \ldots, \varphi_{i-1}^k, \varphi_i^{k-1}, \varphi_{i+1}^{k-1}, \ldots, \varphi_{N^2-1}^{k-1}, \varphi_{N^2}^{k-1} | \boldsymbol{y})}\right]. \quad (5.6)$$

If the proposed value is accepted, then $\varphi_i^k = \varphi_i^*$; otherwise it is rejected and $\varphi_i^k = \varphi_i^{k-1}$. The proposed algorithm is summarised in Figure 5.2. For reliable estimation, this procedure should be repeated until the chain converges to its

Initialise with an arbitrary level-set matrix $\boldsymbol{\varphi}^0$.

Repeat the following steps for $k = 1, \dots, K$

Repeat the following steps for $i = 1, \dots, N$

1. Generate ϵ from a Gaussian distribution $N(0, \kappa_1^2)$.
2. Generate a new value φ_i^*, such that $\varphi_i^* = \varphi_i^k + \epsilon$.
3. Evaluate the acceptance probability $\alpha(\varphi_i^* | \varphi_i^k)$ in Equation 5.6.
4. Generate u from a uniform distribution $U(0, 1)$. If $u \leq \alpha$, the proposed φ_i^* is accepted, and $\varphi_i^k = \varphi_i^*$. Otherwise, φ_i^* is rejected, and $\varphi_i^k = \varphi_i^{k-1}$.

FIGURE 5.2: *The single-component random walk M-H algorithm.*

equilibrium distribution. Thus, it is useful to check Markov chain paths and to calculate sample autocorrelation functions. In equilibrium, the paths should look random and well mixed; in addition, autocorrelation functions should be close to zero for all but small lags.

Once the sample of the level-set matrix $\boldsymbol{\varphi}$ has been generated from the posterior distribution, with only sampled $\boldsymbol{\varphi}$ after a suitable burn-in period considered, then various estimates are available (Aykroyd, 2015). Let $\boldsymbol{\varphi}^1, \boldsymbol{\varphi}^2, \dots, \boldsymbol{\varphi}^K$ be the generated sample collected in equilibrium, then functions of $\boldsymbol{\varphi}$ can be estimated by

$$\widehat{g(\boldsymbol{\varphi})} = \frac{1}{K} \sum_{k=1}^{K} g(\boldsymbol{\varphi}^k).$$

The level-set matrix $\widehat{\boldsymbol{\varphi}}$ is estimated by the pixel-wise posterior mean which is given by

$$\widehat{\varphi}_i = \frac{1}{K} \sum_{k=1}^{K} \varphi_i^k.$$

The generated samples can be used to estimate the domain partition, that is the zero level-set curve along with the number of regions, region intensities and the noise variance. Also, it is possible to estimate the marginal distributions of the parameters using the corresponding sample histograms. The location of the interfaces between regions can be estimated using the zero level-set of the sample functions $\boldsymbol{\varphi}^1, \boldsymbol{\varphi}^2, \ldots, \boldsymbol{\varphi}^K$, which gives the sample $\widehat{\gamma}^1, \widehat{\gamma}^2, \ldots, \widehat{\gamma}^K$. Moreover, for each k the number of object regions can be calculated using function `contourLines` in R. All the zero contour lines are obtained with their locations by applying the function to the level-set matrix $\boldsymbol{\varphi}$; the number of these zero contour lines is the number of object regions.

Further, each of the generated sample $\boldsymbol{\varphi}^1, \boldsymbol{\varphi}^2, \ldots, \boldsymbol{\varphi}^K$ can be used to estimate the corresponding object mean $\hat{\mu}_*^1, \hat{\mu}_*^2, \ldots, \hat{\mu}_*^K$ as follows:

$$\hat{\mu}_*^k = \frac{1}{n_*^k} \sum_{i \in I_*^k} y_i, \qquad \text{for} \qquad 1 \leq k \leq K,$$

where n_*^k is the number of data points in the corresponding region $I_*^k = \{i : \varphi_i^k \geq 0\}$. Also, the corresponding background mean sample $\hat{\mu}_0^1, \hat{\mu}_0^2, \ldots, \hat{\mu}_0^K$ can be estimated as

$$\hat{\mu}_0^k = \frac{1}{n_0^k} \sum_{i \in I_0^k} y_i, \qquad \text{for} \qquad 1 \leq k \leq K,$$

where n_0^k is the number of data points in the corresponding region $I_0^k = \{i : \varphi_i^k < 0\}$. Finally, the noise variance can be estimated as follows:

$$(\hat{\sigma}^2)^k = \frac{1}{N^2 - 1} \left[\sum_{i \in I_0^k} (y_i - \hat{\mu}_0^k)^2 + \sum_{i \in I_*^k} (y_i - \hat{\mu}_*^k)^2 \right], \qquad \text{for} \qquad 1 \leq k \leq K.$$

Then, the mean of the object, the background means, and the noise variance are estimated respectively by

$$\widehat{\mu}_* = \frac{1}{K} \sum_{k=1}^{K} \hat{\mu}_*^k, \qquad \widehat{\mu}_0 = \frac{1}{K} \sum_{k=1}^{K} \hat{\mu}_0^k, \qquad \text{and} \qquad \widehat{\sigma}^2 = \frac{1}{K} \sum_{k=1}^{K} (\hat{\sigma}^2)^k.$$

In addition, it is possible to construct the credible interval (see Besag et al. (1995) and Aykroyd (2015)); for the level-set matrix. This is done by calculating the credible intervals for all components of the level-set matrix. Let $\varphi_i^{(k)}$ be the k^{th} observation in the order sample, then a 95% Bayesian credible interval for the component φ_i can be estimated from the 2.5% and 97.5% sample quantiles of the generated sample, that is

$$\left(\varphi_i^{(0.025K)}, \varphi_i^{(0.975K)}\right).$$

The same calculation is performed at each component φ_i, $i = 1, \ldots, n$, in the level-set matrix. The results can then be used to construct the credible region for the zero level-set curve of the mean level-set matrix, by linking the lower and upper quantiles for the components of the level-set matrix. Similarly, a 95% Bayesian credible interval for an estimated function $g(\boldsymbol{\varphi})$ can be calculated as

$$\left(g(\boldsymbol{\varphi})^{(0.025K)}, g(\boldsymbol{\varphi})^{(0.975K)}\right),$$

where $g(\boldsymbol{\varphi})^{(k)}$ is the k^{th} observation in the ordered sample. Finally, the posterior probability of pixel i being part of an object, where $1 \leq i \leq N^2$, can be estimated using the generated sample $\varphi_i^1, \varphi_i^2, \ldots, \varphi_i^K$ as follows:

$$\widehat{\Pr(i \in I_*)} = \frac{1}{K} \sum_{k=1}^{K} I(\varphi_i^k),$$

where $I(.)$ is an indicator function defined as

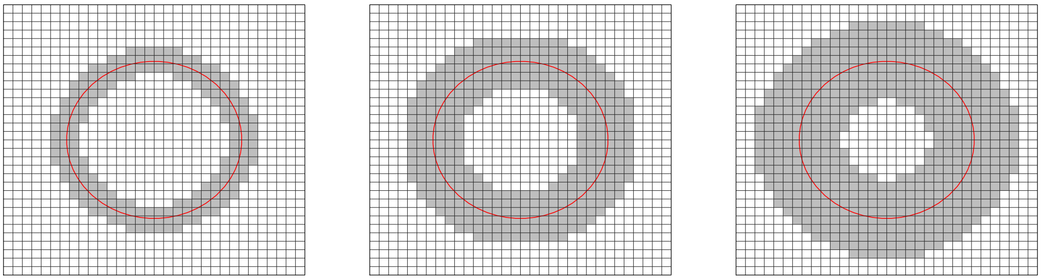
$$I(\varphi_i^k) = \begin{cases} 1 & \text{if} \quad i \in I_*^k, \\ 0 & \text{if} \quad i \in I_0^k. \end{cases} \tag{5.7}$$

5.6 Implementation issues

Narrow-band technique

In the standard LSM, the narrow-band technique is usually used to overcome the problem of excessive computation times for large data problems (Osher and Sethian, 1988). The main idea of the narrow-band technique is to update the level-set matrix only in a narrow band of interest, denoted by B. This is done by considering pixels around the zero level-set contour and ignoring all other pixels.

The proposed M-H algorithm described in the previous section produces samples of the level-set matrix $\boldsymbol{\varphi}$, and is based on single component updates for



(a) $\delta = 0.05$ (b) $\delta = 0.1$ (c) $\delta = 0.15$

FIGURE 5.3: *Narrow-band sizes for different choices of δ.*

all components of the level-set matrix. However, the interest is only in estimating the zero level-set, thus the level-set matrix only needs to be updated for components in regions close to the zero level-set. A new adaptive method is proposed, based on the narrow-band technique, for updating the level-set matrix in the algorithm. Rather than updating all components of the level-set matrix, the focus is only on components which are in a narrow band around the zero level-set. The key step is to update the level-set matrix only at these pixels in the narrow-band. This adaptive technique achieves a significant speed-up while maintaining accurate results.

The width of the narrow band is determined by the tuning parameter δ, which should be chosen carefully to balance the speed of the algorithm and the accuracy of the results. There are two extreme cases. If δ is very small, then there are no components in the narrow band to update, the level-set matrix is not updated, and the zero level-set contour does not change. At the other extreme, if δ is very large, the whole image is in the narrow band, and there is no reason to use the technique as all components are being updated. Figure 5.3 shows the zero level-set (red contour) with the narrow band (grey region) for different values of δ.

In addition, since the prior parameter τ in Equation 5.4 controls the amount of shrinkage, it should be chosen very carefully. Figure 5.4 shows the number of pixels inside the narrow-band region after applying the technique; it is clear that as the value of τ increases the number of pixels inside the narrow-band

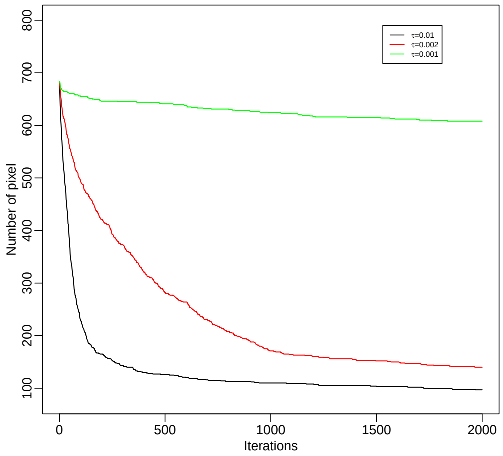
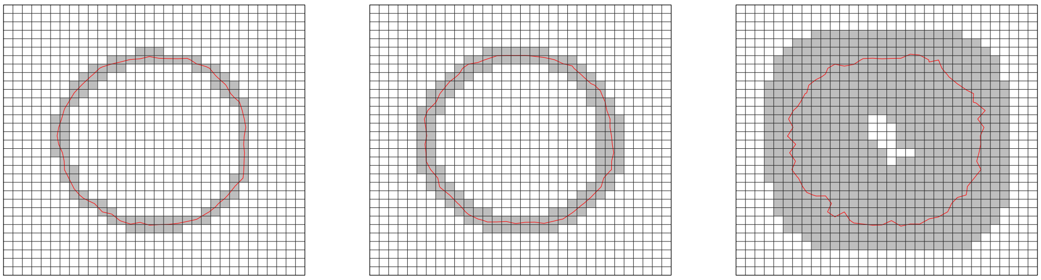


FIGURE 5.4: *Number of pixels inside the narrow-band region for different values of τ.*



(a) $\tau = 0.01$

(b) $\tau = 0.002$

(c) $\tau = 0.001$

FIGURE 5.5: *Narrow-band sizes for different values of τ.*

region decreases. This is because the amount of shrinkage increases as seen in Figure 5.5. Thus there is a link between the best choice of narrow-band size δ and τ; the optimum values of δ and τ should balance the algorithm's speed and convergence.

It is useful for a few iterations in the burn-in period to update all components, as this will prevent disconnected regions from forming. If disconnected regions do form, then the algorithm may not converge to the correct solution, because the same pixels would be updated in each iteration; thus the zero level-set curve will not change. Also when the narrow band technique is used then the number of updated pixels needs to be monitored during the iterations, as the contour might change its shape, which suggests a change to δ as well.

Sample size calculation

An important issue with MCMC algorithms is how many sweeps should be produced after the burn-in period; there is a good discussion of this issue in the lecture notes by Sokal (1989). The MCMC method forms a Markov chain where the generated samples are not independent, thus they will have an asymptotic variance $\text{var}_{asy} = \frac{\tau\sigma^2}{n}$, where $\tau = \sum_{t=-\infty}^{\infty} \rho(t)$ is the integrated autocorrelation time and $\rho(t)$ is the autocorrelation function of the process. It is clear that var_{asy} is larger than in the case where the samples are independent, thus it can be used to estimate the sample size which is needed after the burn-in period (Aykroyd and Mardia, 2003). The sample size can be chosen so that the value of var_{asy} is smaller than, say, 1% of the sampling variance σ^2 (Aykroyd and Green, 1991). Thus n is chosen to satisfy

$$\frac{\text{var}_{asy}}{\sigma^2} = \frac{\tau(\sigma^2/n)}{\sigma^2} = \frac{\tau}{n} < \frac{1}{100}.$$

Let $\rho(t)$ be the autocorrelation function at time t for a function f. Sokal (1989) defines the integrated autocorrelation time as

$$\tau = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_{(t)}.$$

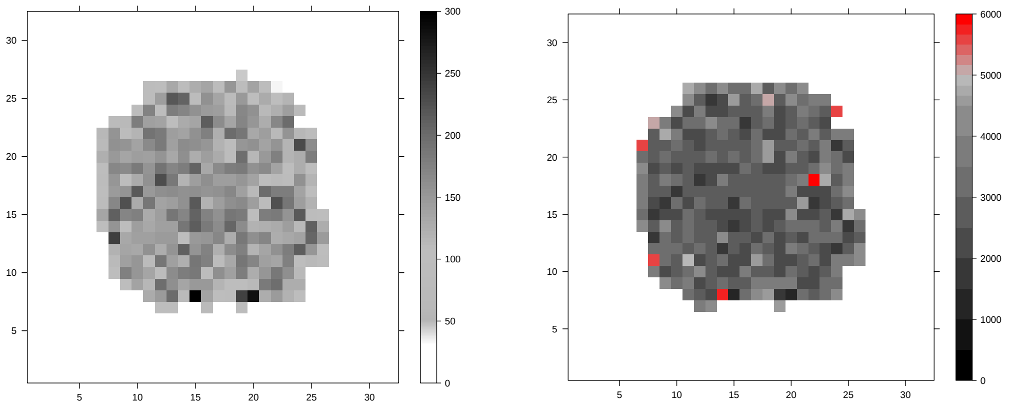
Now let $\bar{f} = \frac{1}{n}\sum_{t=1}^{n} f_t$ be the sample mean which has variance

$$\text{var}(\bar{f}) = \frac{\text{var}(f)}{n}\tau.$$

The variance of the sample mean is larger by a factor τ than it would be if the generated sample values were independent. Using this, the effective sample size is given by

$$n_e = \frac{n}{\tau}.$$

The calculations of the effective sample size are obtained using the Function `effectiveSize` in R. In particular, the function is used to calculate the size n_e for each element φ_i individually. Figure 5.6(a) shows the effective number of independent sample values in a run of length 5000. It is clear that the effective



(a) *The effective sample size*

(b) *The required sample size*

FIGURE 5.6: *Sample size calculation for pixels in the narrow band after burn-in period.*

sample size for all elements in the narrow band is roughly in a range between 100 and 300. In the same manner, the required sample size, n_r, can be calculated following Aykroyd and Mardia (2003), which yields

$$n_r = \frac{n}{n_e} \times 100.$$

The results are shown in Figure 5.6(b). One clearly sees that the number of sample values generated needs to be increased to achieve a reasonable accuracy of the results for a few components (shown by the red colour). However, for most components (shown by the grey colour) the required sample size is below 5000, and hence the run length is acceptable.

5.7 Simulation experiments

In this section simulation experiments have been carried out to identify objects in a given image, i.e. to divide the image into meaningful regions. This is done using the Bayesian model via the M-H algorithm, which is proposed in this chapter to estimate the boundary of the objects through the zero level-set curve. The numerical results are based on a number of synthetic images, which vary in both the number of objects and the shape of the objects. In particular, the simulation experiments can be divided into two parts. In the first part, the simulations are considered as introductory examples to explore the performance of the proposed algorithm. In each simulation an image was generated with one object against the background. In the second part, further exploration of the performance is made by applying the proposed method to simulated images that have multiple regions of contrasting shape and area. In all simulations, the aim is to identify objects in images which are composed of 1024 pixels in a lattice of dimensions 32×32.

The data in all simulation experiments were generated from the data model given by Equation 5.3, where I_0 denotes the background region with mean intensity parameter μ_0, I_* denotes the object regions with mean intensity parameter μ_*, and the common noise variance σ^2. The aim is to estimate both the number of regions and the interfaces between the regions, along with the region intensities and noise variance. All these estimates are based on estimating the level-set function. In order to estimate the values of components of the level-set matrix, the proposed M-H algorithm is used to generate samples of the level-set matrix and, as usual, the M-H output from the iterations in the burn-in period are discarded, to reduce the effect of the initial value. In particular, a main run of 4500 iterations for every example has been taken after a burn-in period of 500 iterations. At the beginning of the burn-in period, all components were updated, but after 100 iterations, the narrow-band technique was applied and only components near the zero level-set contour were updated.

The main advantage of using MCMC methods, in addition to parameter estimation, is to explore the posterior distribution. In particular, the marginal posterior distributions for the object mean μ_*, the area of the objects, and the length of their boundaries are considered, where the area and the length are measured in pixel units. In order to check the convergence, and to estimate the required sample size, the samples generated by the proposed algorithm are graphically and numerically summarized using the trace and autocorrelation plots.

Circle example

In this example, the image is simulated with one circular object located at the centre, the intensities $\mu_* = 5$ and $\mu_0 = 7$ for the object and the background respectively, and Gaussian noise ($\sigma = 0.3$). Figure 5.7(left) shows a cloud of

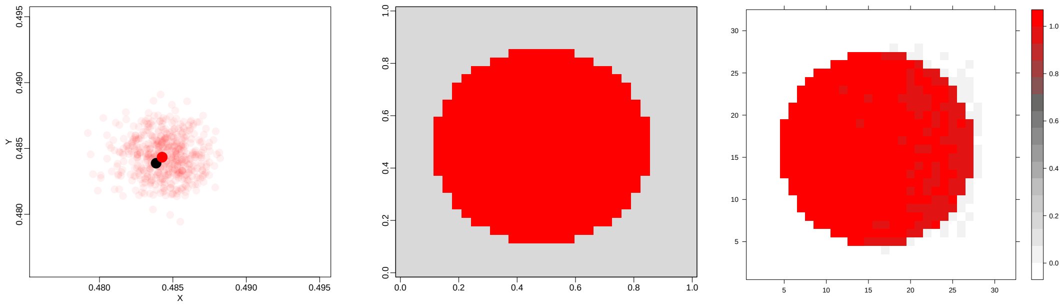


FIGURE 5.7: *A sub-sample of the location of the object centre with the true location (in black) (left), the posterior mean estimate of the image (centre) and the posterior probability of pixel i being part of the object (right).*

points for a sub-sample of the location of the object centre chosen randomly, with the true location (in black). Clearly the variability is small and the true location lies within this cloud of points, which shows the uncertainty in the location of the circle. The posterior estimate of the image is shown in the same figure (centre), from which it is clear that both the true and estimated images are similar. To confirm this result, the correct classification rate was calculated, and no misclassified pixel were found. The posterior probability of each pixel being part of the object is calculated, showing what region, background or object, is most likely to occur in each pixel. The result is shown in Figure 5.7 (right), where the posterior probabilities are close to 1 for all pixels which in this case are known to be from the object region, and close to zero for those known to be from the background. Therefore, it can be said that the accuracy of the object identifying is very good. Table 5.1 shows that the estimates are close to the true values. The posterior standard deviations presented in the table show that the area of the object was estimated more accurately than the other parameters.

Parameter	True value	Post. mean	Post. std dev	Post. skewness	Post. kurtosis
μ_*	5.000	5.006 (4.995 , 5.019)	0.007	0.950	4.719
A_* (pixel)^2	420.500	417.662 (413.274, 422.063)	2.664	0.005	2.944
L_* (pixel)	76.770	77.686 (74.158, 80.392)	1.882	-0.786	6.078

TABLE 5.1: *Summary of posterior results for circle example.*

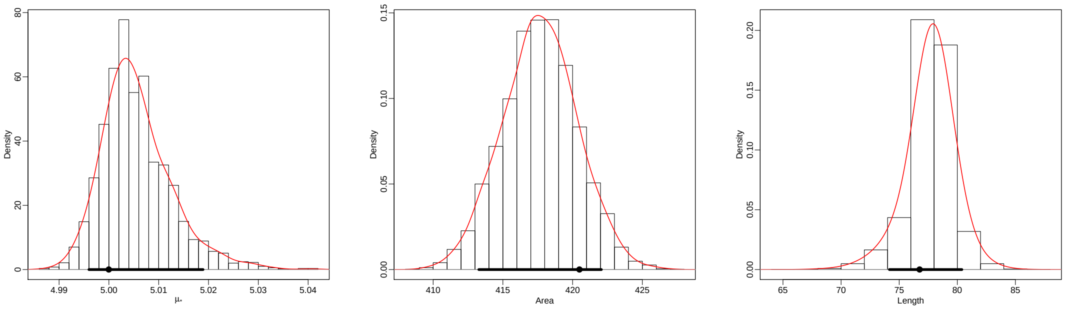


FIGURE 5.8: *Histograms and density estimates, with* 95% *credible intervals for: the object mean* $\hat{\mu}_*$ *(left), the object area (centre) and the length of its boundary (right).*

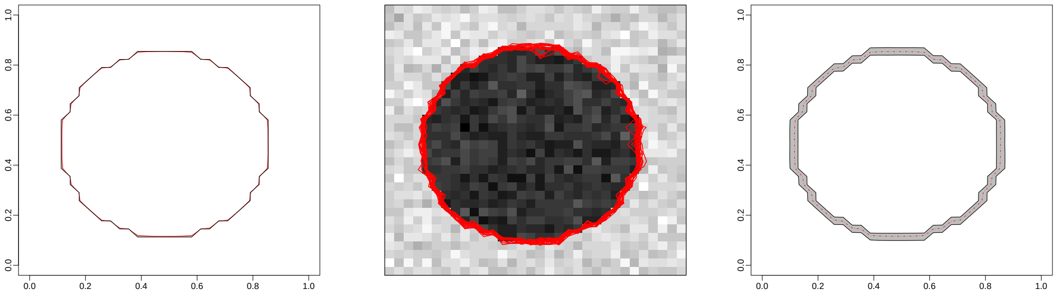


FIGURE 5.9: *Estimation errors of the estimated boundary (left), multiple boundary traces (centre), and the estimated boundary (shown in red), with the* 95% *credible interval of the object boundary (shown in grey) (right).*

Figure 5.8 shows the posterior histograms with density estimates for the object mean μ_* (left), the area of the circular object (centre) and the length of its boundary (right). The thick line on the horizontal axis indicates the 95% credible intervals. Clearly, the small variation in these parameters is showing that the parameters are well-estimated. In addition, the object area has an approximately symmetric distribution; however, the sampling distribution for the object mean μ_* is positively skewed and for the boundary length is negatively skewed. The skewness and kurtosis values presented in Table 5.1 support these findings.

The accuracy in the circle identification is shown in Figure 5.9. The estimation errors (left) are defined as the difference between the true and the estimated boundaries, the thickness of the band indicates the estimation error which are so small that they can not be seen clearly. The estimation variability is also

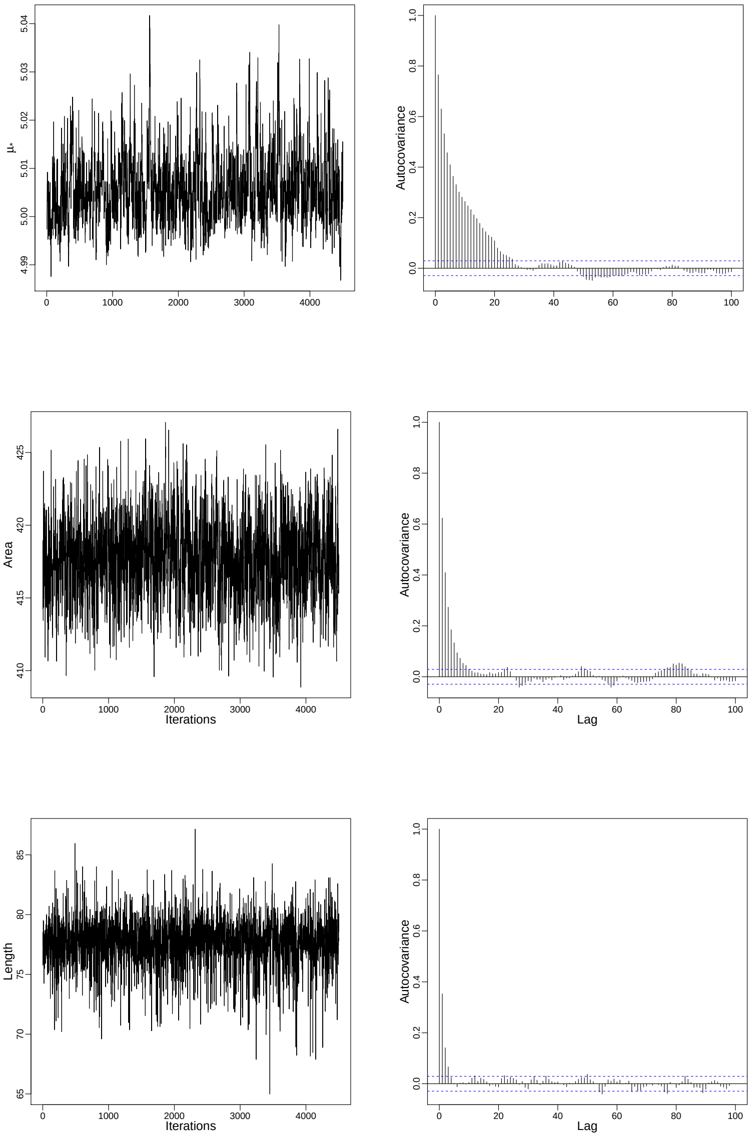


FIGURE 5.10: *Monitoring statistics for the M-H algorithm for segmenting the circular object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean μ_*, the object area, and the length of its boundary.*

shown in the figure, with the central panel showing multiple curves for selected iterations after the convergence of the algorithm. The uncertainty of the boundary location is given by the spread of the sample curves. In particular, a sub-sample of 20 curves is chosen randomly from the sample of 4500. It is clear that the variability is small at the boundaries, which shows the certainty in the location of the estimated boundary. Additionally, the right panel shows the credible interval of the level-set contour obtained, by linking the lower and upper quantiles for the level-set components of the level-set function in the generated sample. The central red dashed curve shows the location of the mean of the level-set contour. It is clear that the thickness of the grey region shows a small and equal variability around the estimated boundary.

The trace plots of the object mean, the object area and the length of its boundary are shown in the left panels of Figure 5.10. Also, the autocorrelation functions for these parameters are shown in the right panels of the same figure. From the trace plots, it is clear that the algorithm reaches equilibrium and the simulation appears to have stabilized. In addition, the autocorrelation functions show that the correlation decays quickly. Therefore, there is evidence that the proposed method has converged to a good solution and there is confidence in the conclusions made from the results.

The choice of prior smoothing parameter β is now discussed. In practice, it is often the case that the smoothing parameter is unknown and needs to be estimated. Here the values of β are selected numerically using a straightforward approach estimating the boundaries with different values of β, and then chooses the value with minimum MSE. Figure 5.11 shows the results from the posterior distribution using different values of β. It is clear that, when $\beta = 0.1$, the sample boundaries are more widely spread around the posterior mean of the zero level-set curve. In addition, the difference between the estimated zero level-set curve and the true boundary is quite large. For smaller β values $\beta = 0.05$ and $\beta = 0.01$, the sample boundaries become more regular around the posterior mean of the zero level-set curve, with each sample curve becoming more circular, and the difference between the estimated zero level-set curve and the true boundary being small. The preferred value of the prior smoothing parameter used here (and in the following experiments) is $\beta = 0.01$.

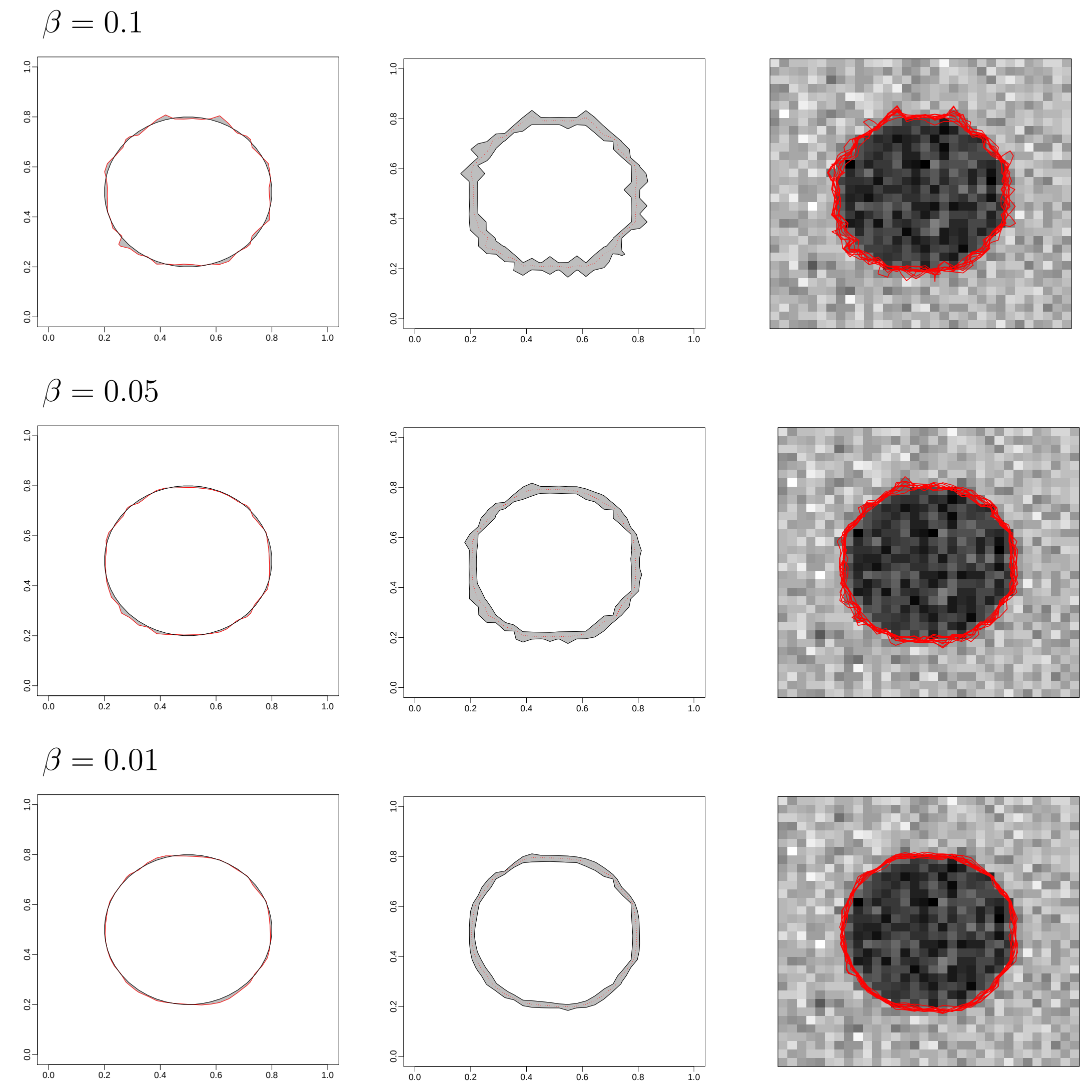


FIGURE 5.11: *Multiple zero level-set* γ showing the effect of changing the prior parameter β. The values of β are: $\beta = 0.1$ (top row), $\beta = 0.05$ (middle row), and $\beta = 0.01$ (bottom row).

Square example

In this example, the results are based on simulated data, with a square object located at the centre of the image. The data have the intensities $\mu_* = 5$ and $\mu_0 = 7$ for the object and the background respectively, and the Gaussian noise ($\sigma = 0.3$). A cloud of points for a sub-sample of the location of the object centre chosen randomly is shown in Figure 5.12 (left). It is clear that the variability around the mean centre is small and the true location lies within this cloud of points, this indicates the certainty in the location of the square. In the central panel, the posterior mean of the image is shown with correct classification rate

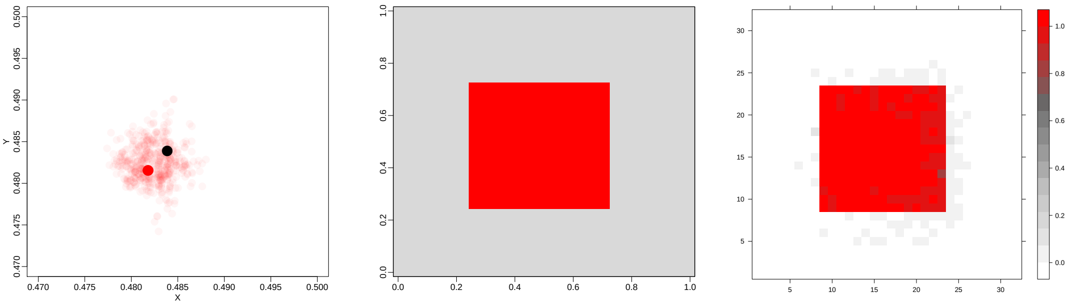


FIGURE 5.12: *A sub-sample of the location of the object centre with the true location (in black) (left), the posterior mean estimate of the image (centre) and the posterior probability of pixel i being part of the object (right).*

equal to 1. Thus, the true and estimated images are identical. The right panel shows the posterior probability of the pixel being part of the object. It is clear that about one object pixel has a low probability to be from the object region, however the other object pixels have high probabilities to be from the square region. Table 5.2 shows that the estimates are relatively close to the true values. The posterior standard deviations presented in the table also show that the area of the object was estimated more accurately than the other parameters.

Parameter	True value	Post. mean	Post. std dev	Post. skewness	Post. kurtosis
μ_*	5.000	4.999	0.008	1.098	1.779
		(4.989 , 5.015)			
A_*	224.500	223.809	2.797	0.296	0.255
(pixel)^2		(219.421, 228.544)			
L_*	59.933	62.461	2.744	0.039	1.376
(pixel)		(57.701, 67.129)			

TABLE 5.2: *Summary of posterior results for square example.*

Histograms and density estimates are shown in Figure 5.13, for the object mean μ_* (left), the area of the object (centre), and the length of its boundary (right), along with the 95% credible intervals of these estimates. All histograms show the same variability as in the circle example, and the parameters are estimated quite well. The length of the boundary has an approximately symmetric distribution with skewness value close to zero. In contrast, the object mean and the

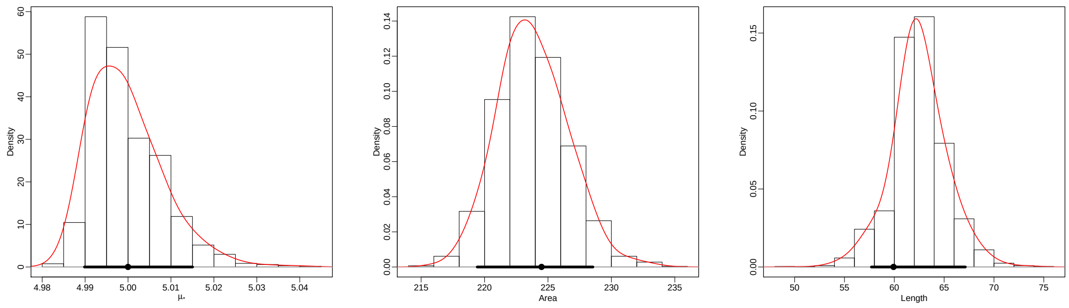


FIGURE 5.13: *Histograms and density estimates, with* 90% *credible intervals for: the object mean* $\hat{\mu}_*$ *(left), the object area (centre), and the length of its boundary (right).*

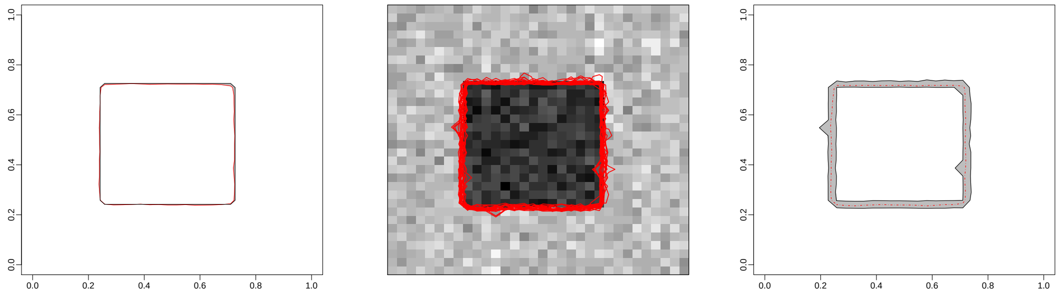


FIGURE 5.14: *Estimation errors of the estimated boundary (left), multiple boundary traces (centre), and the estimated boundary (shown in red), with the* 95% *credible interval of the object boundary (shown in grey) (right).*

area of the object have asymmetric distributions with positive skewness. The skewness and kurtosis values are presented in Table 5.2.

Figure 5.14 shows the accuracy and the variability in the square segmentation. The left panel shows small errors, with a slightly greater difference between the true and the estimated boundaries at the top-right corner. A random sub-sample of 20 curves is shown in the same figure (centre), showing small variability, and certainty in the location of the estimated boundary. The right panel shows the credible interval of the level-set contour, where the central red dashed curve shows the location of the mean of the level-set contour. It can be said that the proposed algorithm has been successful in identifying the square object, although the variation is greater at the top-left and bottom-right than elsewhere.

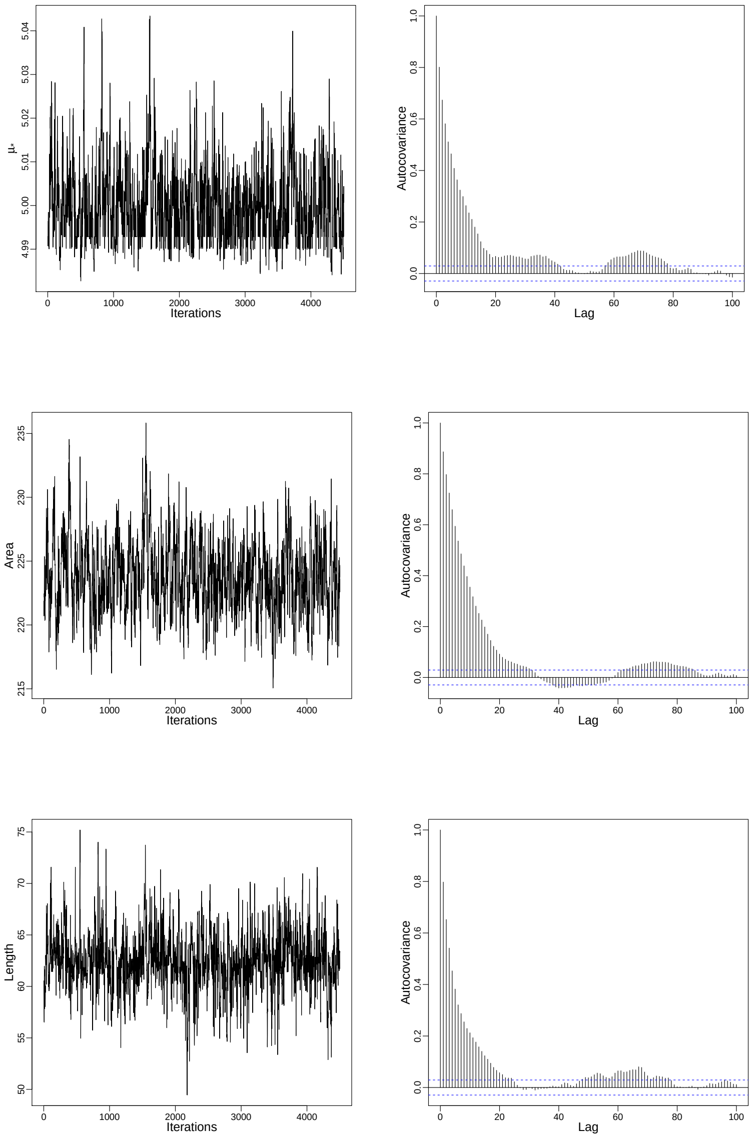


FIGURE 5.15: *Monitoring statistics for the M-H algorithm for segmenting the square object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean μ_*, the object area, and the length of its boundary.*

From the trace plots shown in the left panels of Figure 5.15, it is clear that the simulation appears to have stabilized. Also, from the autocorrelation functions presented in the right panels of the same figure, it is clear that the correlation decays quickly. Thus, it can be said that there is no cause for concern when using the results in the parameter estimation.

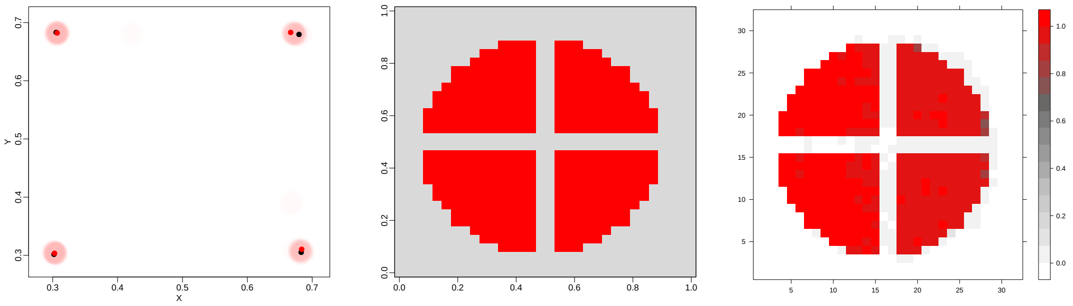


FIGURE 5.16: *Sub-samples of the location of the centres with the true locations (in black) (left), the posterior mean estimate of the image (centre) and the posterior probability of pixel i being part of the object (right).*

Cut-circle example

To further explore the proposed algorithm, the true object geometry considered here is a cut-circle shape, located at the centre of the image and consisting of four equal parts. Thus, the aim is to fit the proposed model including unknown boundaries of the four identical objects. The data is simulated with the intensities $\mu_* = 5$ and $\mu_0 = 7$ for the object and the background respectively, and Gaussian noise ($\sigma = 0.3$). Figure 5.16(left) shows cloud of points for sub-samples of the location of the object centres chosen randomly, with the true locations (in black). The variability is small and the true locations lie within these clouds of points, clearly this indicates the certainty in the location of the cut-circle object. The posterior mean of the image is shown in the same figure (centre). It is clear that both the true and estimated images are similar and, unsurprisingly, when the correct classification rate was calculated, there were no misclassified pixels. The posterior probability of the pixel being part of the object is calculated and the result is shown in the right panel. It is quite clear that the posterior probabilities for most object pixels are roughly 1, with a few object pixels less than 0.8. Therefore, it can be said that the accuracy of the object identification is very good. Table 5.3 shows that the estimates for the object mean and the boundary length are closer to the true values than the estimate for the object area. However, the posterior standard deviations presented in the table show that the area of the object was estimated more accurately than the other parameters.

Figure 5.17 shows estimates of posterior distributions as histograms and as

Parameter	True value	Post. mean	Post. std dev	Post. skewness	Post. kurtosis
μ_*	5.000	5.005	0.007	1.234	2.167
		(4.996 , 5.018)			
A_*	408	402.860	4.703	-0.043	-0.034
(pixel)^2		(395.030, 410.505)			
L_*	166.255	170.431	3.343	-0.455	1.079
(pixel)		(165.826,176.517)			

TABLE 5.3: *Summary of posterior results for cut-circle example.*

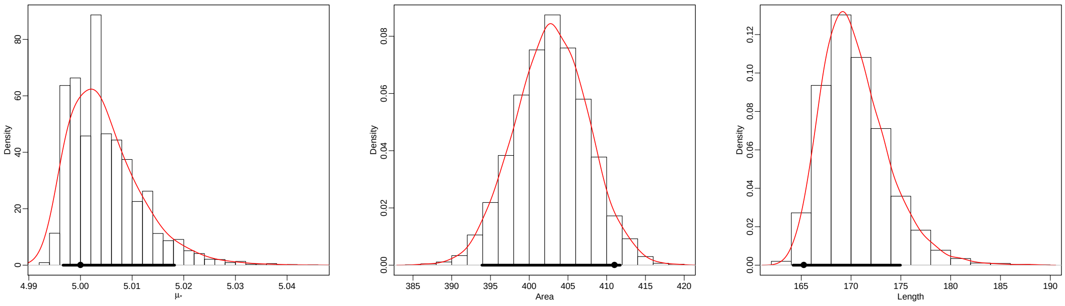


FIGURE 5.17: *Histograms and density estimates, with* 90% *credible intervals for: the object mean* $\hat{\mu}_*$ *(left), the object area (centre), and the length of its boundary (right).*

smoothed density estimates for the object mean μ_* (left), the area of the cut-circle object (centre) and the length of its boundaries (right). The thick line on the horizontal axis indicates the 95% credible interval. It is clear from the figure that the area of the cut-circle object has an approximately symmetric distribution. However, the object mean and the length of the boundaries, again, have positive-skew and negative-skew distributions, respectively. The posterior summary of these parameters is presented in Table 5.3.

The accuracy and the variability in the cut-circle identification is shown in Figure 5.18. The errors in the left panel are quite small; however, the estimated errors are greater in the corners, especially on the right. The central panel shows a random sub-sample of 20 curves from the sample of 4500. One can see that the variability is small at all the boundaries, which shows the certainty in the location of the estimated boundary. Finally, the credible interval of the estimated boundary can be seen in the right panel, where the central red dashed curves show the location of the estimated boundary. Clearly, the figure shows

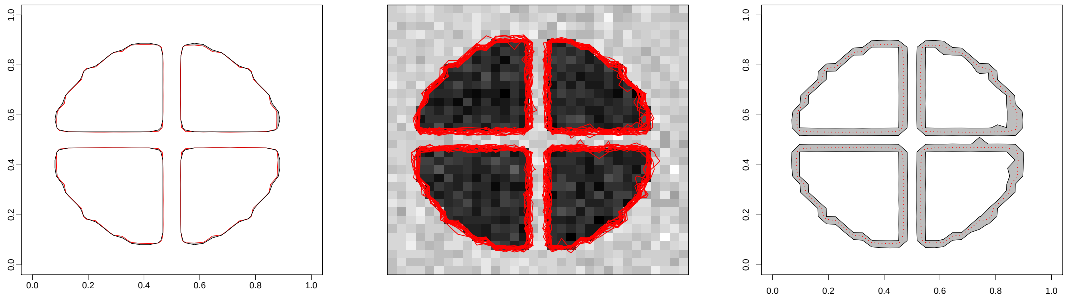


FIGURE 5.18: *Estimation errors of the estimated zero level-set contour (left), multiple boundary traces (centre), and the estimated boundaries (shown in red), with the* 95% *credible interval of the boundaries (shown in grey) (right).*

that the greatest variability around the estimated boundary is in the quarters on the right. Although the cut-circle object is considered as a challenge in image segmentation, the proposed algorithm has identified the boundaries of the four parts of the cut-circle successfully.

The trace plots of the object mean, the object area and the length of its boundary are shown in the left panels of Figure 5.19, from which it is clear that the simulations appear to have stabilized. Also, the autocorrelation functions for these parameters are shown in the right panels of the same figure, with the plots showing that the correlation decays quickly. Thus, it can be said that convergence has been reached, and that the results are reliable.

In general for MCMC methods, moving from one solution to another can be difficult. However, the proposed algorithm here moves from one solution to the other easily and quickly. Figure 5.20 shows some selected iterations. Here, it can be seen there are some iterations in which the level-set matrix φ in the generated sample defines one zero level-set contour, with a single corresponding object region; for other iterations, there are two defined zero level-set contours, with two object regions. The figure indicates good mixing between one and two object regions.

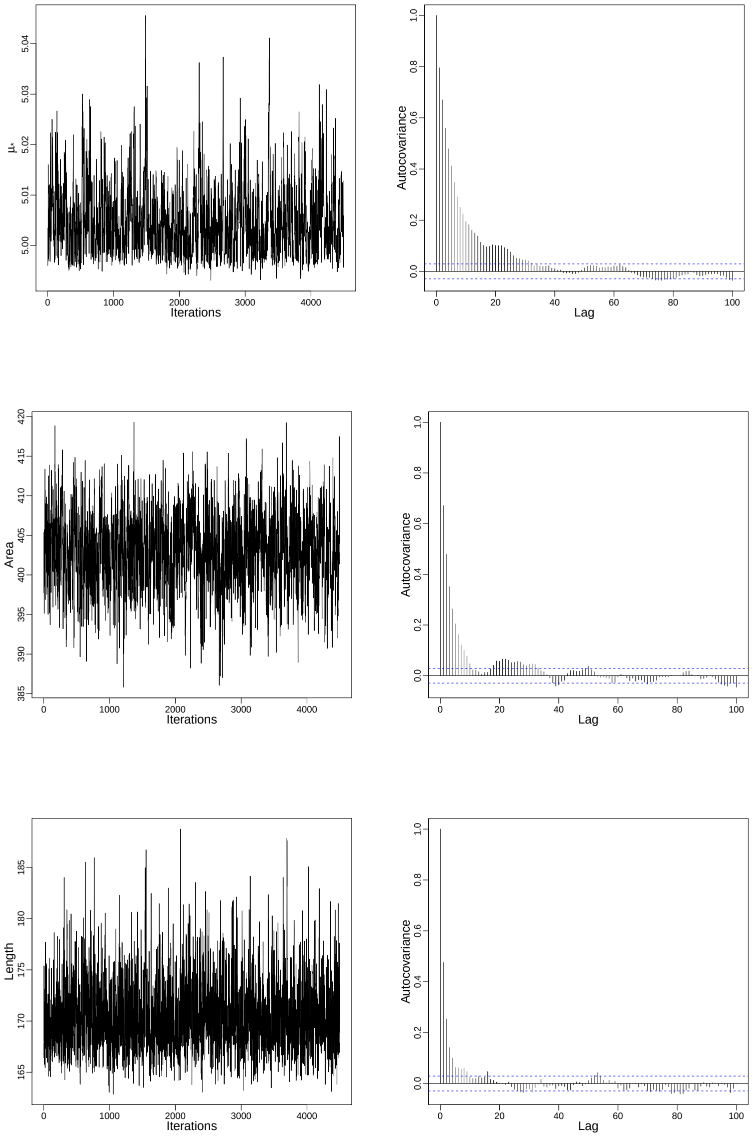


FIGURE 5.19: *Monitoring statistics for the M-H algorithm for segmenting the cut-circle object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean μ_*, the object area, and the length of its boundary.*

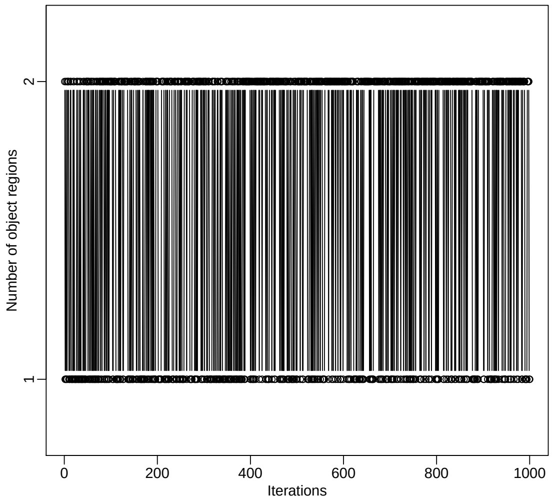


FIGURE 5.20: *Number of object regions across iterations.*

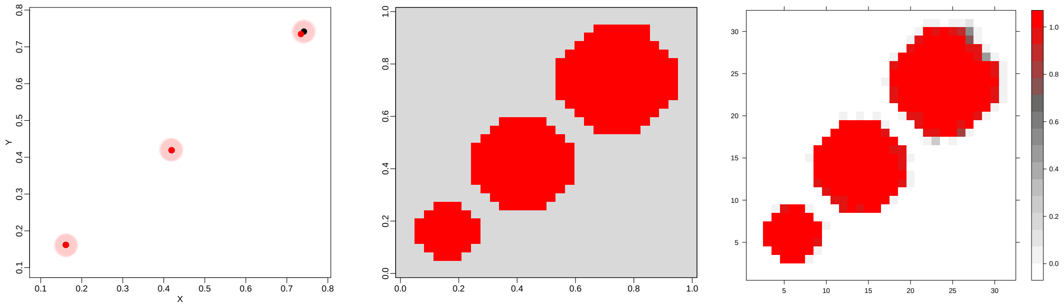


FIGURE 5.21: *Sub-samples of the location of the object centres with the true locations (in black) (left), the posterior mean estimate of the image (centre) and the posterior probability of pixel i being part of the object (right).*

Three-circle example

In this example, the image consists of three circular objects with different sizes. The data is simulated with the intensities $\mu_* = 5$ and $\mu_0 = 7$ for the object and the background respectively, and the Gaussian noise ($\sigma = 0.4$). Samples of the locations of the object centres are chosen randomly and shown in Figure 5.21(left) with the true locations (in black). It is clear that for each sub-sample, the variation occurs around the mean centre is small and the true location lies within this sample, which indicates the certainty in the object location. The posterior mean estimate of the image is shown in the same figure (centre), where

Parameter	True value	Post. mean	Post. std dev	Post. skewness	Post. kurtosis
μ_*	5.000	5.014	0.013	0.676	1.021
		(4.996 , 5.034)			
A_*	261.500	256.463	3.559	0.140	0.062
(pixel)^2		(250.814, 262.370)			
L_*	99.397	105.004	4.879	0.083	0.325
(pixel)		(97.167,113.268)			

TABLE 5.4: *Summary of posterior results for three-circle example.*

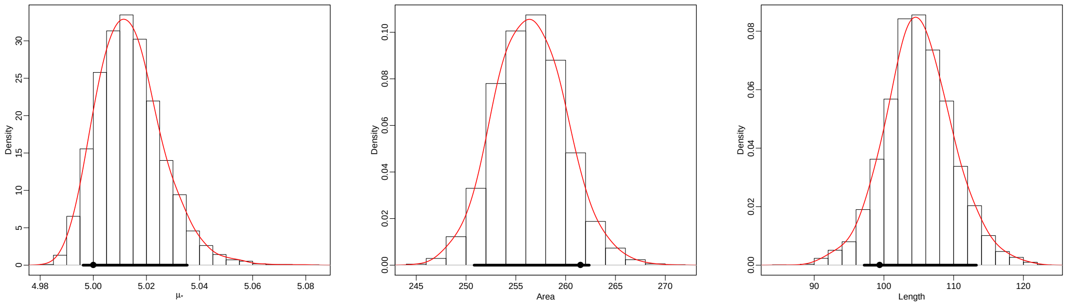


FIGURE 5.22: *Histograms and density estimates, with* 90% *credible intervals for: the object mean* $\hat{\mu}_*$ *(left), the object area (centre), and the length of its boundary (right).*

the correct classification rate is 0.998. The posterior probability of the pixels being part of the objects is calculated and the result is shown in the right panel. It is quite clear that the posterior probabilities for most object pixels are roughly 1, with a few object pixels being between 0.4 and 0.6. Table 5.4 shows that the estimates are relatively close to the true values. The posterior standard deviations presented in the table show that the area of the object was estimated more accurately than the other parameters.

Figure 5.22 shows the posterior histograms with density estimates for the object mean μ_* (left), the area of the objects (centre), and the length of their boundaries (right). The thick line on the horizontal axis indicates the 95% credible intervals. Clearly, the small variation in these parameters is showing that they are well-estimated. In addition, the sampling distributions for the object mean and the area of the objects are positively skewed, whilst the distribution for the length of the boundaries is symmetric, with the skewness value close to zero. The numerical results are presented in Table 5.4.

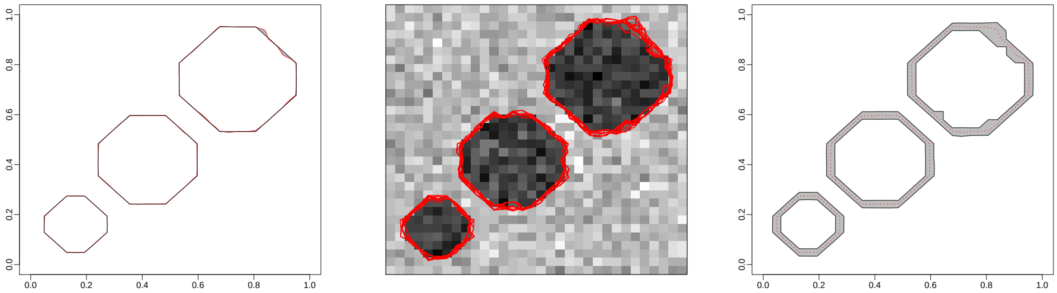


FIGURE 5.23: *Estimation errors of the estimated boundaries (left), multiple boundary traces (centre), and the estimated boundaries (shown in red), with the* 95% *credible interval of the boundaries (shown in grey) (right).*

The accuracy in the identification of the objects is shown in Figure 5.23. The estimation errors (left) are defined as the differences between the true boundaries and the estimated boundaries, the thickness of which indicates the level of estimation errors. It is clear that the errors are quite small in general, however the estimated errors are considerable in the top-right of the biggest circle. The estimation variability is also shown in the figure, where the central panel shows multiple contours for some selected iterations, with the uncertainty of the boundary location given by the spread of the sample curves. In particular, a sub-sample of 20 boundaries of each object is chosen randomly from the sample of 4500. It is clear that the variability is small at the boundaries, which shows the certainty in the location of the estimated boundaries. Also, the right panel shows the credible interval of the estimated boundaries, which indicates that there is greater variability at the top-right of the biggest circle than elsewhere.

The trace plots of the mean intensity of the objects, the area of the objects, and the length of their boundaries are shown in the left panels of Figure 5.24. In addition, the autocorrelation functions for these parameters are shown in the right panels of the same figure. From the trace plots, it is clear that the algorithm reaches equilibrium and the simulation appears to have stabilized. Furthermore, the autocorrelation functions show that the correlation decays quickly. Therefore, there is evidence that the proposed method has converged to an acceptable solution, and there is confidence in the conclusion made from the results.

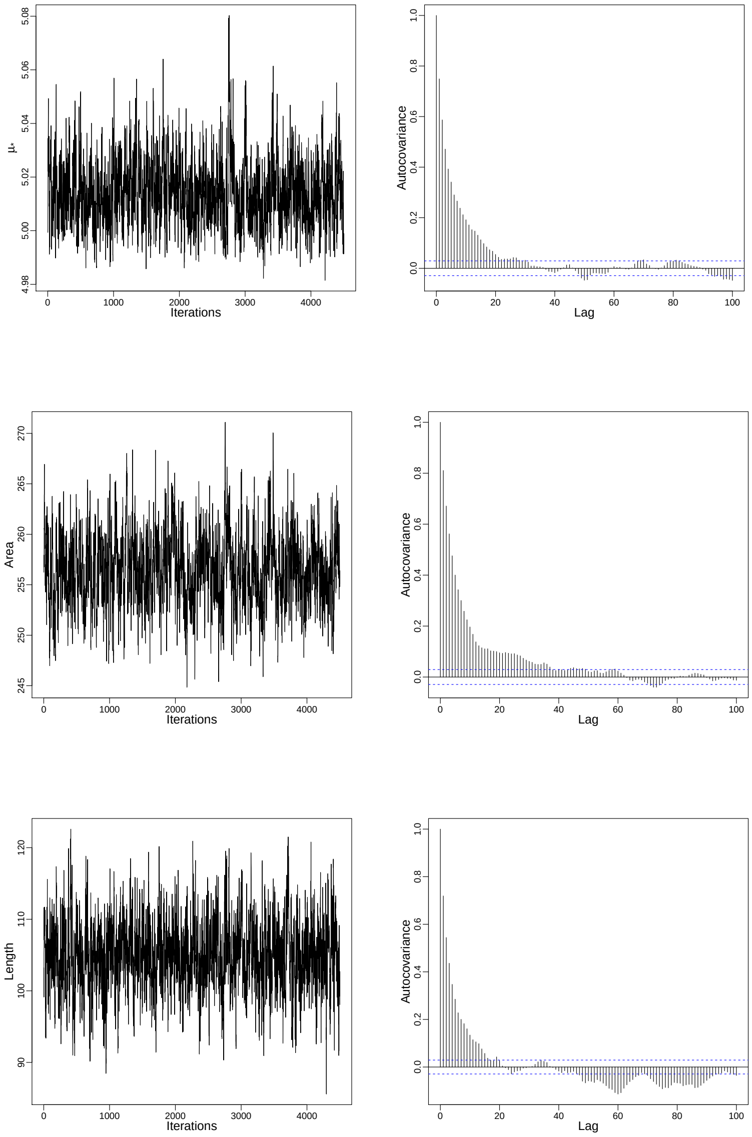


FIGURE 5.24: *Monitoring statistics for the M-H algorithm for segmenting the circular object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean μ_*, the object area, and the length of its boundary.*

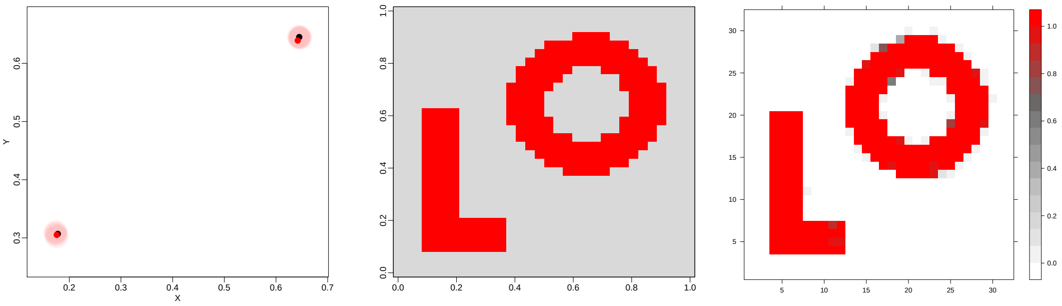


FIGURE 5.25: *Sub-samples of the location of the object centres with the true locations (in black) (left), the posterior mean estimate of the image (centre), and the posterior probability of pixel i being part of the object (right).*

L & O example

Parameter	True value	Post. mean	Post. std dev	Post. skewness	Post. kurtosis
μ_*	5.000	5.014	0.009	0.577	3.427
		(4.999 , 5.031)			
A_*	355.500	359.701	2.772	-0.034	3.177
(pixel)^2		(355.357, 364.433)			
L_*	75.213	75.954	1.251	-0.096	3.402
(pixel)		(73.957,77.994)			

TABLE 5.5: *Summary of posterior results for L & O example.*

The objects in this example are L and O shaped objects. The data is simulated with the intensities $\mu_* = 5$ and $\mu_0 = 7$ for the objects and the background respectively, and the Gaussian noise $\sigma = 0.4$. Figure 5.25 shows clouds of points for sub-samples of the locations of the object centres chosen randomly, with the true location (in black). Clearly the variability is small, and the true locations lie within these clouds of points, which shows the certainty in the location of the objects. The posterior mean estimate of the image is shown in the same figure (centre), where the correct classification rate is 0.999. The result of the posterior probability of the pixels being part of the object is shown in the right panel. It is quite clear that the posterior probabilities for most object pixels are roughly 1, with a few object pixels around 0.6. Table 5.5 shows that the estimates are close to the true values. Also, the posterior standard deviations presented in the table show that the area of the object was estimated more accurately than the other

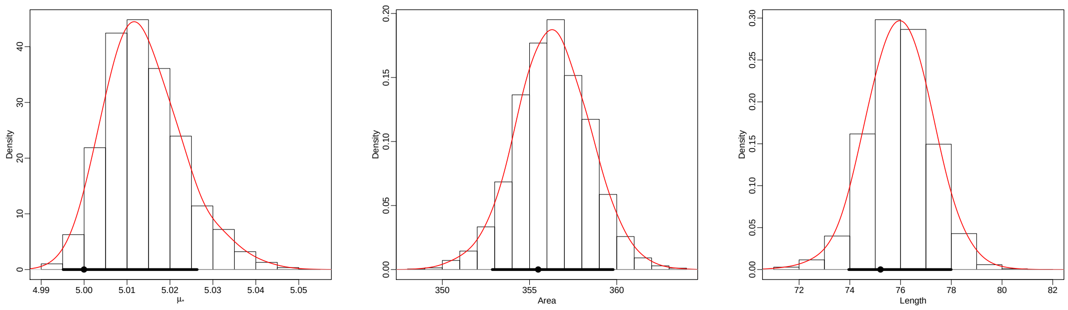


FIGURE 5.26: *Histograms and density estimates, with 90% credible intervals for: the object mean $\hat{\mu}_*$ (left), the object area (centre) and the length of its boundary (right).*

parameters.

Figure 5.26 shows the posterior histograms with density estimates for the mean of the objects μ_* (left), the area of the objects (centre), and the length of their boundaries (right). The thick line on the horizontal axis indicates the 95% credible intervals. Clearly, the small variation in these parameters shows that the parameters can be estimated well. In addition, the sampling distributions for the area of the objects and the length of their boundaries are symmetric, whereas the distribution of the mean of the objects, μ_*, is positivity skewed. The skewness and kurtosis values are presented in Table 5.5.

The trace plots of the mean intensity of the objects, the area of the objects, and the length of their boundaries are shown in the left panels of Figure 5.27. Also, the autocorrelation functions for these parameters are shown in the right panels of the same figure. From the trace plots, it is clear that the algorithm reaches equilibrium, and the simulation appears to have stabilized. In addition, the autocorrelation functions show that the correlation decays quickly. Therefore, the proposed method has converged sufficiently, and there is a confidence in the conclusion made from the results.

The accuracy in the identification of the objects is shown in Figure 5.28. The left panel shows that the errors are quite small; however, it should be noted that the estimated errors are considerable in the top-left of the inner and outer boundaries of the O object, and also in the corners of the L object. The estimation variability is also shown in the figure, where the central panel shows multiple boundaries for selected iterations. It is clear that the variability is small

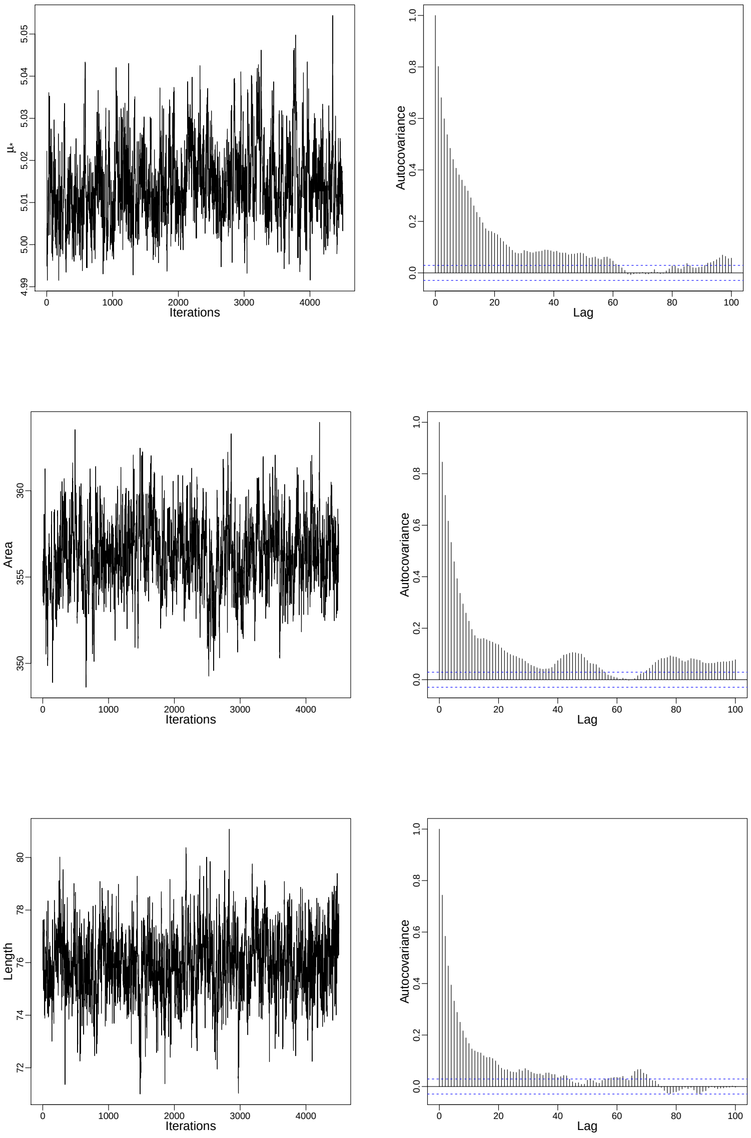


FIGURE 5.27: *Monitoring statistics for the M-H algorithm for segmenting the circular object: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the object mean μ_*, the object area, and the length of its boundary.*

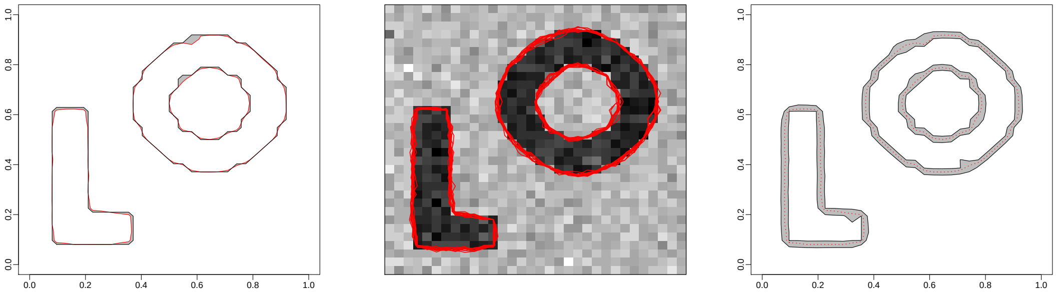


FIGURE 5.28: *Estimation errors of the estimated boundaries (left), multiple boundary traces (centre), and the estimated boundaries (shown in red), with the* 95% *credible interval of boundaries (shown in grey) (right).*

at the boundaries, which shows the certainty in the location of the estimated boundaries. Finally, the right panel shows the credible interval of the estimated boundaries, which indicate that there is greater variability at the top-left of the inner and outer boundaries of the O object, and also at the right-side of the L object.

5.8 Application to real data

In this section, the proposed algorithm is applied to real data. A lot of scientific and clinical research has focused on detecting tumors and designing new therapies. An accurate tissue diagnosis can be obtained through needle biopsies, by the surgical pathologist. In pathology studies, the analysis of a biopsy from human tissue provides information for detection and treatment of deadly diseases, such as cancer (Wang et al., 2011). Histology is the study of the microscopic structure of tissues. The visual interpretation of the cell nucleus structure is used to distinguish normal tissue from cancer tissue. In doing this, objects are detected; this means obtaining the location of the objects (near the object centroid) without identifying the boundaries. The detection technique can help in counting, tracking, and segmentation of the nucleus cells (Xing and Yang, 2016). The identification of the objects is required to distinguish the different structures in the cell, and to isolate those containing biologically significant structures. In particular, the aim in analyzing fluorescence microscopy images of cells is to identify the cell objects and segment their nuclei; this analysis gives important diagnostic information in some tissue lesions, because the nuclei in

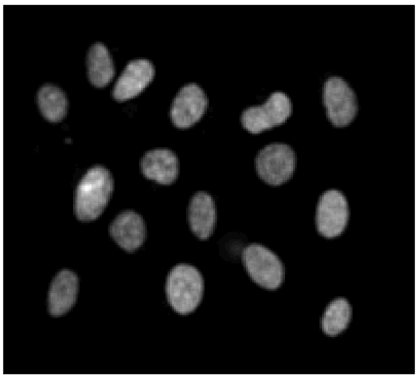


FIGURE 5.29: *The image of real data showing U2OS nuclei cells.*

tumors have characteristic appearances and features such as size, shape, membrane contours, etc. (Wang et al., 2011). Thus, the segmentation approaches are applied to microscopy images of cells from different samples to obtain comprehensive descriptions of the cells.

There are some challenges to cell image segmentation, including, for example, inhomogeneous illumination across the visual field, variation in object shape, size and orientation, and variation of the intensity of objects from the same type, caused for example by inconsistent staining (Nattkemper, 2004). However, many methods have been proposed for the segmentation of cell nuclei in fluorescence microscopy images, including methods based on the Chan–Vese level-set model (e.g. Bergeest and Rohr (2012), Gharipour and Liew (2016), and Zhang and Li (2017)). An overview on segmentation of micrograph images and the problem of segmentation evaluation can be found in Nattkemper (2004). A comprehensive review of the recent nucleus and cell segmentation approaches can be found in Xing and Yang (2016).

The proposed approach is applied to a 2D fluorescence microscopy image of U2OS cell nuclei from (Coelho et al., 2009). The U2OS cells are human osteosarcoma cells (a malignant tumor of bone). The cell line was established in 1964 and was cultivated from tibia tissue of a 15-year-old human female suffering from bone osteosarcoma (Niforou et al., 2008). The data image consists of 16 nuclei, and with a size of 1030×1349 pixels. A large image size will increase the performance of the proposed algorithm, however it will also increase the time

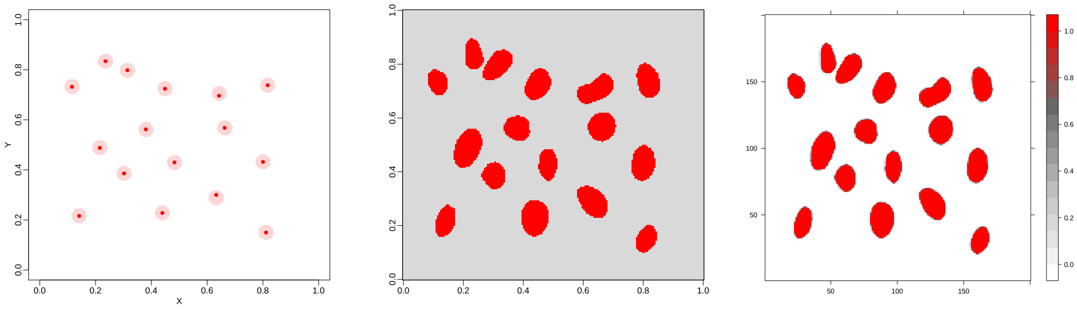


FIGURE 5.30: *Sub-samples of the location of the object centres (left), the posterior mean estimate of the image (centre), and the posterior probability of pixels being part of the objects (right).*

cost. Considering both factors (the segmentation performance and the computational cost), the size of the image was reduced to 200×200 pixels.

Figure 5.30(left) shows sub-samples of the locations of the cell centres chosen randomly. It is clear that for each sub-sample the dispersion around the mean of the centre location is small, which indicates the certainty in the location of the cells. The posterior mean estimate of the image is shown in Figure 5.30(centre), while the estimates of the posterior probabilities of the pixel being part of the objects is shown in Figure 5.30(right). The posterior probabilities can be seen to be close to 1 for all object pixels, however for a few pixels on the boundaries they are close to 0.6. Numerical results are summarised in Table 5.6, from which it it is clear that all parameters were estimated accurately, with small values of the posterior standard deviations and reasonably narrow intervals. Therefore, it can be said that the accuracy of identifying the objects is very good.

Parameter	Post. mean	Credible interval	Post. skewness	Post. kurtosis
μ_*	0.1634	(0.1630, 0.1638)	-0.0163	3.0539
A_* (pixel)^2	112.9391	(112.4929, 113.3955)	0.0362	2.9053
L_* (pixel)	164.1863	(163.3548, 165.0638)	0.1647	3.1998

TABLE 5.6: *Summary of posterior results for real data.*

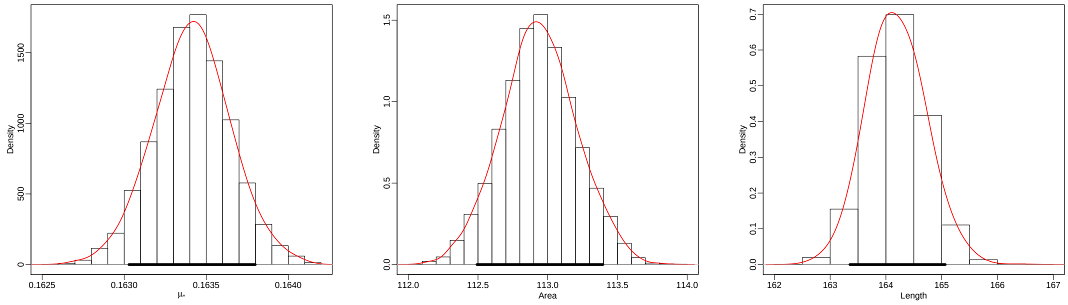


FIGURE 5.31: *Marginal posterior samples generated by the proposed method summarised using histograms and kernel density curves with* 95% *credible intervals for: the object mean* $\hat{\mu}_*$ *(left), the area of objects (centre), and the length of boundaries (right).*

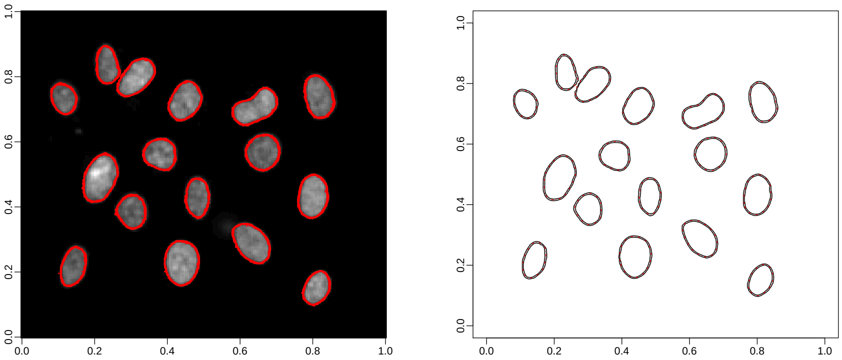


FIGURE 5.32: *Multiple boundary traces (left), and the estimated boundaries (shown in red), with the* 95% *credible interval of the boundaries (shown in grey) (right).*

Figure 5.31 shows the posterior histograms with density estimates for the mean of the objects μ_* (left), the area of the objects (centre), and the length of their boundaries (right). The thick line on the horizontal axis indicates the 95% credible intervals. Clearly, the small variation in these parameters shows that the parameters are well-estimated. Furthermore, the sampling distributions, for the mean intensity of the objects, μ_*, and the area of the objects, are close to symmetric, whereas the distribution of the length of the boundaries is positivity skewed. The skewness and kurtosis values are presented in Table 5.6.

The accuracy and the variability in the object identification is shown in Figure 5.32. The left panel shows multiple boundaries for selected iterations. It is clear

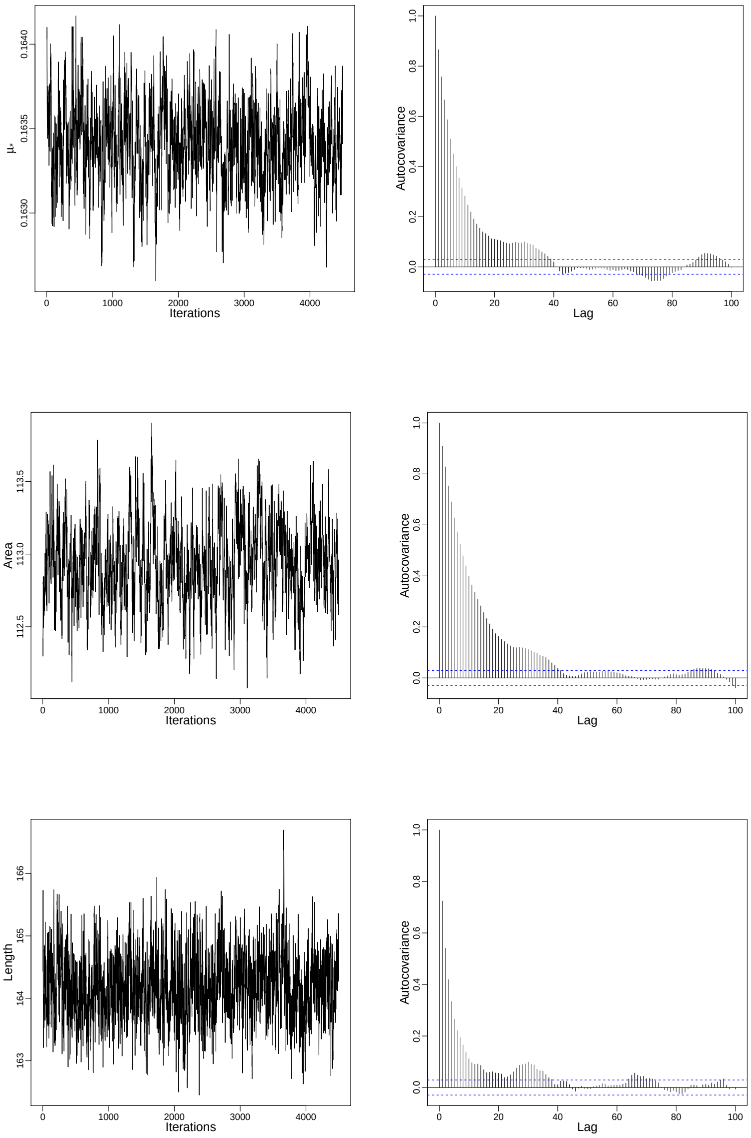


FIGURE 5.33: *Monitoring statistics for the M-H algorithm for the real data segmentation: posterior traces (left column), and the corresponding autocorrelation functions (right column) of the mean μ_*, the area of the objects area, and the length of boundaries.*

that the variability is quite small at the boundaries, which shows the certainty in the location of the estimated boundaries. Finally, the right panel shows minimum variations around all object boundaries.

The trace plots of the mean of the objects, the area of the objects, and the length of their boundaries are shown in the left panels of Figure 5.33. Also, the autocorrelation functions for these parameters are shown in the right panels of the same figure. From the trace plots, it is clear that the Markov chain paths show convergence with good random fluctuations. In addition, the autocorrelation functions show that the correlation decays quickly.

5.9 Conclusions

This chapter has proposed a new approach to image segmentation, through level sets. The level-set matrix was estimated, in a fully Bayesian framework, using a Markov chain Monte Carlo algorithm. This Bayesian formulation incorporates prior information about the level-set matrix. The flexibility of the MCMC approach allows to demonstrate a range of output summaries. The sample mean was used to estimate the posterior mean of the boundary, and sample percentiles to estimate confidence bounds. Marginal distributions, and functions of the simulated values were inspected. The approach has been verified in a range of simulated images as well as real image. The results clearly indicate that the combined Bayesian and MCMC procedure has worked well, and that the algorithms provides a very good and fast estimation to the objects.

Chapter 6

Temporal modelling of the level-set matrix

6.1 Introduction

Finding objects in a given image is one of the most important tasks in image analysis, where the number of objects, their locations, sizes, and shapes may be unknown. With dynamic images, the aim is to track moving objects in a sequence of images. Here the objects evolve and move with time in a coherent manner, where locations, sizes, and shapes might change. For example, the geometries of the objects might change with shrinking and expanding regions, causing a change in the number of the regions. It has been seen in the early chapters that the level-set method can cope with changes in shapes easily and quickly. The idea in this chapter is to extend the level-set method to the case of segmenting moving objects in a sequence of images by including motion information as a criterion for segmentation of a sequence of images. The motion estimation based on the level-set method has been investigated by several researchers (see for example Cremers and Soatto (2003), Cremers and Soatto (2005) and references therein). Also a review can be found in Cremers et al. (2007) and Mitiche and Aggarwal (2016).

In this chapter, the aim is to model an evolving process sequentially based on a sequence of images. In particular, a temporal model is proposed in a Bayesian framework for level-set based image sequence segmentation. Markov chain

Monte Carlo (MCMC) methods are used to explore the model and to obtain information about solution behaviour. The proposed model includes temporal prior information; thus it is able to describe the relations between temporal changes, making identification of the objects more efficient and reducing the estimation variability. There are many applications of similar approach in different areas such as medical imaging (West et al., 2004), industrial process monitoring (West et al., 2005), and geophysics (Uhlemann et al., 2015).

The next section gives a mathematical description of the proposed model and the full details of how the temporal information can be incorporated in the model. The use of the M–H algorithm to estimate the model parameters is described in Section 6.3. Section 6.4 shows the results from the implementation of the algorithm on some simulated image sequences, Section 6.5 includes some conclusions.

6.2 Bayesian Modelling

6.2.1 Likelihood

Assume that a two-dimensional domain $\mathbf{S} \subset \mathbb{R}^2$ is divided into an $N \times N$ grid of equal-sized square pixels, where N is an integer number. Pixel i is located at s_i, where $1 \leq i \leq N^2$; then, the vector of these locations $\mathbf{s}$ forms a finite regular square lattice. Now, define a discrete variable $x_i \in \mathbb{R}$ for each pixel i. Thus the unknown intensities can be given by the vector $\mathbf{x} = \{x_1, \ldots, x_{N^2}\}$ and can be shown as a grey-level image. Now consider a sequence of images which are collected at T time points $\{t_1, \ldots, \ldots, t_T\}$. At each time point t_j, $1 \leq j \leq T$, a finite set of measurements $\mathbf{y}^{t_j} = \{y_1^{t_j}, \ldots, y_{N^2}^{t_j}\}$ is taken at the locations s_i, where $1 \leq i \leq N^2$, and the data can be displayed as a grey-level image. Then, the observed data $\mathbf{y}^{t_j}$ are collected to give the full data set $\mathbf{y} = \{\mathbf{y}^{t_1}, \ldots, \mathbf{y}^{t_T}\} \in \mathbb{R}^{N^2} \times T$.

Now, for each time point t_j, $1 \leq j \leq T$, the domain $\mathbf{S}$ is partitioned into m^{t_j} non-overlapping regions, $\mathbf{R}^{t_j} = \{R_0^{t_j}, \ldots, R_{(m^{t_j}-1)}^{t_j}\}$, with the corresponding intensities $\boldsymbol{\mu}^{t_j} = \{\mu_0^{t_j}, \ldots, \mu_{(m^{t_j}-1)}^{t_j}\}$. Hence, the domain $\mathbf{S}$ can be represented as $R_0^{t_j} \cup \ldots \cup R_{(m^{t_j}-1)}^{t_j} = \mathbf{S}$ with $R_j^{t_j} \cap R_l^{t_j} = \emptyset$, for $j \neq l$. Further $\mathbf{R}_*^{t_j} = \{R_1^{t_j}, \ldots, R_{(m^{t_j}-1)}^{t_j}\}$ denotes all objects, and $R_0^{t_j}$ denotes the background in

S. Thus, $\mathbf{R}_*^{t_j} = (R_0^{t_j})^c$, with corresponding intensities $\boldsymbol{\mu}_*^{t_j} = \{\mu_1^{t_j}, \ldots, \mu_{(m^{t_j}-1)}^{t_j}\}$. In the level-set formulation, the boundaries between the different regions $\mathbf{R}^{t_j}$ are given by the zero level-set γ^{t_j} of the level-set matrix $\boldsymbol{\varphi}^{t_j}$. The data $\mathbf{y}^{t_j}$ depend on the parameter set $\boldsymbol{\theta}^{t_j} = \{\boldsymbol{\varphi}^{t_j}\}$ through a mapping to the corresponding intensity function $\mathbf{x}(\boldsymbol{\theta}^{t_j})$; hence the measurement model is

$$\mathbf{y}^{t_j} = \mathbf{x}(\boldsymbol{\theta}^{t_j}) + \boldsymbol{\epsilon}, \tag{6.1}$$

where $\boldsymbol{\epsilon}$ is a vector of independent and identically distributed Gaussian errors. Thus the conditional distribution of the measurements $\mathbf{y}^{t_j}$ given the parameter set $\boldsymbol{\theta}^{t_j}$ is $\mathbf{y}^{t_j} \mid \boldsymbol{\theta}^{t_j} \sim N(\boldsymbol{\theta}^{t_j}, \sigma^2 I)$ with the likelihood function

$$\begin{aligned} f(\mathbf{y}^{t_j}|\boldsymbol{\theta}^{t_j}) &= \frac{1}{(2\pi\sigma^2)^{N^2/2}} \exp\Bigg\{ -\frac{1}{2\sigma^2} \sum_i (y_i^{t_j} - x_i(\boldsymbol{\theta}^{t_j}))^2 \Bigg\}, \quad \sigma > 0, \\ &= \frac{1}{(2\pi\sigma^2)^{N^2/2}} \exp\Bigg\{ -\frac{1}{2\sigma^2} \Bigg[\sum_{i \in I_0^{t_j}} (y_i^{t_j} - \mu_0)^2 + \sum_{i \in I_*^{t_j}} (y_i^{t_j} - \mu_*)^2 \Bigg] \Bigg\}, \\ &\qquad \sigma > 0, \end{aligned} \tag{6.2}$$

where $I_0^{t_j} = \{i : \varphi^{t_j}(s_i) < 0\}$ and $I_*^{t_j} = \{i : \varphi^{t_j}(s_i) > 0\}$. In the image sequence, each frame is composed of background and objects, where the objects evolve with time. Thus the aim at each time point is to identify the unknown objects; that is, to estimate the boundaries of the objects. Furthermore, the objects might change location, size, and shape; those changes will also be estimated as outputs of the estimation procedure. The changes in the objects with time can be described using a transformation of the boundaries, which include the effects of translation, scaling, and rotation. These transformation effects can be explained in terms of the level-set matrix as follows: two level-set matrices are considered similar if they can be translated, scaled, and rotated to have the same zero level-set. The translation effects are described as shifts in the x- and y-directions with parameters $\mathcal{T}_h$ and $\mathcal{T}_v$, the scale effect is described as shift in the z-direction of the level-set matrix with parameter $\mathcal{T}_z$, and the rotation effect with the angle parameter $\mathcal{T}_r$. Let $\boldsymbol{\mathcal{T}}$ be a vector consisting of parameters involved in the transformation effects. Therefore, the complete parameter set at time t is $\boldsymbol{\theta}^{t_j} = \{\boldsymbol{\varphi}^{t_j}, \boldsymbol{\mathcal{T}}^{t_j}\} = \{\boldsymbol{\varphi}^{t_j}, \mathcal{T}_h^{t_j}, \mathcal{T}_v^{t_j}, \mathcal{T}_z^{t_j}, \mathcal{T}_r^{t_j}\}$. The parameters $\boldsymbol{\mu}$ and σ are unknown, however they can be calculated when $\mathbf{y}^{t_j}$ and $\boldsymbol{\varphi}^{t_j}$ are given.

Clearly it is possible to identify objects in each frame separately using the algorithm proposed in Chapter 5. However, a temporally evolving approach using the temporal prior models described below will be proposed for time points $t_2, \ldots, t_T$.

6.2.2 Prior models

The prior for the level-set matrix $\boldsymbol{\varphi}^{t_j}$ is presented as discussed in Chapter 5, hence the shrinkage and smoothing prior with known parameters τ and β is given by

$$p(\boldsymbol{\varphi}^{t_j}) \propto \exp\left\{ -\frac{1}{2\tau^2}\sum_i (\varphi_i^{t_j})^2 - \frac{1}{2\beta^2}\sum_{i\sim j}(\varphi_i^{t_j} - \varphi_j^{t_j})^2 \right\}, \qquad i = 1, \ldots, T. \quad (6.3)$$

The proposed algorithm can be used to obtain an estimate for the model parameter; that is the level-set matrix given only the observed data. However, the estimation algorithm is extended to include temporal prior information; this is to consider the temporal correlation between the level-set matrices from two consecutive time points. Hence, the level-set matrix $\boldsymbol{\varphi}^{t_j}$ for each time is estimated given the corresponding data $\mathbf{y}^{t_j}$ and the estimated level-set matrix from the previous time point $\boldsymbol{\varphi}^{t_{(j-1)}}$. In particular, in this chapter a temporally evolving segmentation algorithm is proposed using the temporal prior model described as follows.

It is assumed that as the process evolves through time, the changes within the image vary slowly and smoothly. This can be modelled using a temporal prior. Let $\Psi(\boldsymbol{\mathcal{T}}, \boldsymbol{\varphi})$ denote the transformation of the level-set matrix $\boldsymbol{\varphi}$ by the parameter vector $\boldsymbol{\mathcal{T}}$, thus $\Psi(\boldsymbol{\mathcal{T}}, \boldsymbol{\varphi})$ is a translation of $\boldsymbol{\varphi}$ by $\mathcal{T}_h$ and $\mathcal{T}_v$, a scaling by $\mathcal{T}_z$, and a rotation by angle $\mathcal{T}_r$. Then, the temporal prior information is that, given $\boldsymbol{\mathcal{T}}$, the object in the current image will be similar to the object in the previous image. Here, a Gaussian distribution on first-order differences is used to model the temporal smoothness. The temporal prior is then presented as

$$p(\boldsymbol{\varphi}^{t_j}, \boldsymbol{\mathcal{T}}^{t_j} | \boldsymbol{\varphi}^{t_{(j-1)}}) \propto \frac{1}{(2\pi\omega^2)^{N^2/2}} \exp\left\{ -\frac{1}{2\omega^2} \|\boldsymbol{\varphi}^{t_j} - \Psi(\boldsymbol{\mathcal{T}}^{t_j}, \boldsymbol{\varphi}^{t_{(j-1)}})\|^2 \right\},$$
$$i = 2, \dots, T, \quad (6.4)$$

with temporal variance ω^2 controlling the variability in the temporal smoothness. In this model, it is assumed that there is no prior information about the transformation parameters, thus each is distributed uniformly on a finite interval such that the zero level-set curves remain inside the area of the image. In the Bayesian approach the likelihood and the prior are combined to give the posterior distribution of the model parameters $\boldsymbol{\theta}^{t_j}$ given data $\mathbf{y}^{t_j}$ and parameter set $\boldsymbol{\theta}^{t_{(j-1)}}$,

$$\pi(\boldsymbol{\theta}^{t_j} \mid \mathbf{y}^{t_j}, \boldsymbol{\theta}^{t_{(j-1)}}) = \frac{f(\mathbf{y}^{t_j} | \boldsymbol{\theta}^{t_j}) p(\boldsymbol{\theta}^{t_j} \mid \boldsymbol{\theta}^{t_{(j-1)}})}{f(\mathbf{y}^{t_j})}, \qquad i = 2, \dots, T.$$

The estimation of all the unknown model parameters and other posterior inference are based on this posterior distribution. This will be done through Markov chain Monte Carlo methods which will be described in the next section.

6.3 Parameter estimation

The parameters at time $t_2, \dots, t_T$ are given by $\boldsymbol{\theta}^{t_j} = \{\boldsymbol{\varphi}^{t_j}, \mathcal{T}_h^{t_j}, \mathcal{T}_v^{t_j}, \mathcal{T}_z^{t_j}, \mathcal{T}_r^{t_j}\}$. It should be noted that at time t_1 there is no temporal prior information. In general, the estimation approach starts from an arbitrary starting value denoted as $\boldsymbol{\theta}^0$, then a Markov chain is simulated to produce values $\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^K$. The algorithm proposed here is to consider changes to a single parameter at each iteration where the proposal is a perturbation of the current parameter with variance value chosen to achieve reasonable acceptance rates. The general structure of the estimation procedure is similar to the algorithm explained in Chapter 5.

To simplify the notation, consider that the level-set matrix at time t_j is denoted by $\boldsymbol{\varphi} = \{\varphi_1, \dots, \varphi_{N^2}\}$ and the estimated level-set matrix at time $t_{(j-1)}$ is denoted by $\boldsymbol{\psi} = \{\psi_1, \dots, \psi_{N^2}\}$. Also the model parameter vector at time t_j is

$\boldsymbol{\theta} = \{\boldsymbol{\varphi}, \mathcal{T}_h, \mathcal{T}_v, \mathcal{T}_z, \mathcal{T}_r\}$. At iteration k of the MCMC algorithm this will be denoted by $\boldsymbol{\theta}^k = \{\boldsymbol{\varphi}^k, \mathcal{T}_h^k, \mathcal{T}_v^k, \mathcal{T}_z^k, \mathcal{T}_r^k\}$. Here, the estimation approach is essentially the same for all time points. The parameter set $\boldsymbol{\theta}$ is composed of the level-set matrix $\boldsymbol{\varphi}$, $\mathcal{T}_h$, $\mathcal{T}_v$, $\mathcal{T}_z$, and $\mathcal{T}_r$, hence in the estimation procedure, each parameter will be considered separately. The algorithm is now explained and is also summarized in Figure 6.1.

The level-set matrix update: The proposal procedure for the level-set matrix $\boldsymbol{\varphi}$ is based on separate single component updates as described in Chapter 5. For $i = 1, \dots, N^2$, repeat the following steps: a new value for φ_i is proposed, such that $\varphi_i^* = \varphi_i^{k-1} + \epsilon$, with $\epsilon \sim N(0, \kappa_\varphi^2)$. The variance κ_φ^2 is chosen to achieve a reasonable acceptance rate. The proposed value φ_i^* is accepted and the parameter component updated accordingly with probability

$$\begin{aligned}\alpha(\varphi_i^* \mid \varphi_i^{k-1}) \;&=\; \min\left[1, \frac{\pi(\varphi_1^k, \dots, \varphi_{i-1}^k, \varphi_i^*, \varphi_{i+1}^{k-1}, \dots,}{\pi(\varphi_1^k, \dots, \varphi_{i-1}^k, \varphi_i^{k-1}, \varphi_{i+1}^{k-1}, \dots,}\right. \\ &\qquad \left.\frac{\dots, \varphi_N^{k-1}, \mathcal{T}_h^{k-1}, \mathcal{T}_v^{k-1}, \mathcal{T}_z^{k-1}, \mathcal{T}_r^{k-1} \mid \mathbf{y}^{t_j}, \boldsymbol{\theta}^{t_{(j-1)}})}{\dots, \varphi_N^{k-1}, \mathcal{T}_h^{k-1}, \mathcal{T}_v^{k-1}, \mathcal{T}_z^{k-1}, \mathcal{T}_r^{k-1} \mid \mathbf{y}^{t_j}, \boldsymbol{\theta}^{t_{(j-1)}})}\right]. \qquad (6.5)\end{aligned}$$

Then, generate u from a uniform distribution, $U(0, 1)$. If $\alpha(\varphi_i^* \mid \varphi_i^{k-1}) > u$, accept the proposal, and set $\varphi_i^k = \varphi_i^*$, otherwise it is rejected and no change is made; $\varphi_i^k = \varphi_i^{k-1}$.

Shift in x-direction parameter update: A new value $\mathcal{T}_h^*$ is proposed as a perturbation of the value $\mathcal{T}_h^{k-1}$, $\mathcal{T}_h^* = \mathcal{T}_h^{k-1} + \epsilon_h$, where $\epsilon_h \sim N(0, \kappa_h^2)$. The variance κ_h^2 is chosen to achieve a reasonable acceptance rate. Using the parameter values $\boldsymbol{\mathcal{T}}_h^* = \{\mathcal{T}_h^*, \mathcal{T}_v^{k-1}, \mathcal{T}_z^{k-1}, \mathcal{T}_r^{k-1}\}$, the effects of translation, scaling, and rotation are incorporated to calculate the transformation of $\boldsymbol{\psi}$, $\Psi(\boldsymbol{\mathcal{T}}_h^*, \boldsymbol{\psi})$, by evaluating the acceptance probability

$$\alpha(\mathcal{T}_h^* \mid \mathcal{T}_h^{k-1}) = \min\left[1, \frac{\pi(\boldsymbol{\varphi}^k, \mathcal{T}_h^*, \mathcal{T}_v^{k-1}, \mathcal{T}_z^{k-1}, \mathcal{T}_r^{k-1} \mid \mathbf{y}^{t_j}, \boldsymbol{\theta}^{t_{(j-1)}})}{\pi(\boldsymbol{\varphi}^k, \mathcal{T}_h^{k-1}, \mathcal{T}_v^{k-1}, \mathcal{T}_z^{k-1}, \mathcal{T}_r^{k-1} \mid \mathbf{y}^{t_j}, \boldsymbol{\theta}^{t_{(j-1)}})}\right].$$

Then, generate u from a uniform distribution, $U(0, 1)$. If $\alpha(\mathcal{T}_h^* \mid \mathcal{T}_h^{k-1}) > u$, accept the proposal, and set $\mathcal{T}_h^k = \mathcal{T}_h^*$; otherwise $\mathcal{T}_h^k = \mathcal{T}_h^{k-1}$.

Shift in y-direction, and z-direction parameter update: following the above steps for updating $\mathcal{T}_h$.

Angle rotation parameter update: A new value $\mathcal{T}_r^*$ is proposed for the angle rotation parameter $\mathcal{T}_r^{k-1}$, such that $\mathcal{T}_r^* = \mod(\mathcal{T}_r^{k-1} + \epsilon_r, 2\pi)$, where $\epsilon_r \sim N(0, \kappa_r^2)$. The variance κ_r^2 is chosen to achieve a reasonable acceptance rate. Using the parameter values $\boldsymbol{\mathcal{T}}_r^* = \{\mathcal{T}_h^k, \mathcal{T}_v^k, \mathcal{T}_z^k, \mathcal{T}_r^*\}$, the effects of translation, scaling, and rotation are incorporated to calculate the transformation of $\boldsymbol{\psi}$, $\Psi(\boldsymbol{\mathcal{T}}_r^*, \boldsymbol{\psi})$, by evaluating the acceptance probability

$$\alpha(\mathcal{T}_r^* \mid \mathcal{T}_r^{k-1}) = \min\left[1, \frac{\pi(\boldsymbol{\varphi}^k, \mathcal{T}_h^k, \mathcal{T}_v^k, \mathcal{T}_z^k, \mathcal{T}_r^* \mid \mathbf{y}^{t_j}, \boldsymbol{\theta}^{t_{(j-1)}})}{\pi(\boldsymbol{\varphi}^k, \mathcal{T}_h^k, \mathcal{T}_v^k, \mathcal{T}_z^k, \mathcal{T}_r^{k-1} \mid \mathbf{y}^{t_j}, \boldsymbol{\theta}^{t_{(j-1)}})}\right].$$

Then, generate u from a uniform distribution, $U(0,1)$. If $\alpha(\mathcal{T}_r^* \mid \mathcal{T}_r^{k-1}) > u$, accept the proposal, and set $\mathcal{T}_r^k = \mathcal{T}_r^*$; otherwise $\mathcal{T}_r^k = \mathcal{T}_r^{k-1}$.

For reliable estimation, this procedure should be repeated until the chain converges to its equilibrium distribution. Thus, it is useful to check Markov chain paths and to calculate sample autocorrelation functions. In equilibrium, the paths should look unstructured and well mixed; in addition, autocorrelation functions should be close to zero for all but small lags. Once the sample has been generated from the posterior distribution, and collected after a suitable burn-in period to allow the chain to reach equilibrium, the sample mean and variance are used to estimate the posterior mean and variance. In addition, sample order statistics are used to estimate credible intervals, marginal distributions estimated and functions of the parameter values can be studied as well. Various estimates, using the simulated samples of the level-set matrix, were described in Chapter 5. Similarly, point and interval estimates are available from the samples of the temporal parameters. For a particular parameter $\mathcal{T}$ in the transformation vector $\boldsymbol{\mathcal{T}}$; let $\mathcal{T}^1, \ldots, \mathcal{T}^K$ be the simulated sample collected after equilibrium has been reached, then the posterior mean and variance can be estimated by the corresponding sample mean and variance:

$$\widehat{\mathcal{T}} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{T}^k, \qquad \widehat{\sigma}^2 = \frac{1}{K-1}\sum_{k=1}^{K}(\mathcal{T}^k - \widehat{\mathcal{T}})^2.$$

In addition, the $100(1-\alpha)\%$ credible interval for $\mathcal{T}$ is given by

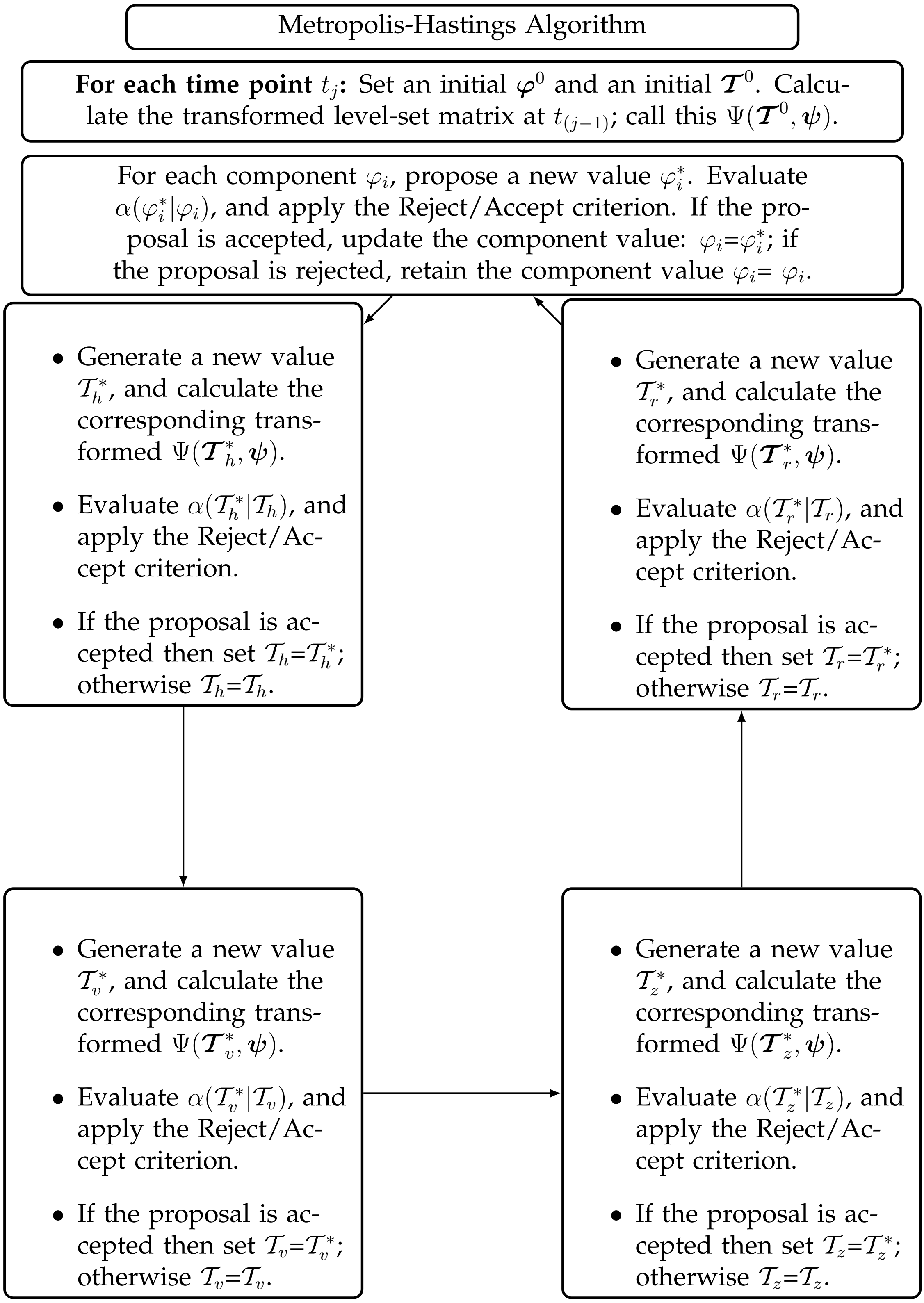
$$\left(\mathcal{T}^{(K(\alpha/2))}, \mathcal{T}^{(K((1-\alpha)/2))}\right),$$

where K is chosen so that $K(\alpha/2)$ and $K((1-\alpha)/2)$ are both integer.

6.4 Simulation studies

In this section two simulation experiments have been carried out to study the estimation of moving objects in a sequence of images, and to estimate the changes with time using the model and algorithm proposed in this chapter. The first simulation acts as an introductory example to explore the performance of the proposed algorithm. In this experiment, a sequence of images were simulated with a single object, and the object evolves and moves slowly and smoothly with time. Here only the results when the temporal priors are included will be considered. In the second experiment, a sequence of images were simulated with two objects moving with time in a smooth way, at the beginning one object obscures the other, then both objects move in such a way that the covered object gradually becomes visible, and at the end the two objects are well separated. In this experiment the details of the full analysis will be given, so in addition to estimating the model parameters, the effect of including the temporal information will be studied.

The simulation experiments consider a fixed image size of 32×32 which is composed of 1024 pixels, and T time points. For each frame, the data were generated from the data model given by Equation (5.3), where I_0 denotes the background region with mean intensity parameter μ_0, I_* denotes the object regions with mean intensity parameter μ_*, and the common noise variance σ^2. In these experiments one is interested in estimating the boundaries of the objects and the four parameters which specify the changes in location, size, and shape over time. The procedure starts at frame 1, in which the boundaries of the objects are estimated using image data $\boldsymbol{y}^1$ only. Then, for frames 2 to T the boundaries of the objects are estimated using the estimate of $\boldsymbol{\varphi}^{t_{(j-1)}}$ along with data $\boldsymbol{y}^{t_j}$. In addition, the other parameters $\mathcal{T}_h^{t_j}$, $\mathcal{T}_v^{t_j}$, $\mathcal{T}_z^{t_j}$, and $\mathcal{T}_r^{t_j}$ are estimated.



Metropolis-Hastings Algorithm

For each time point t_j**:** Set an initial $\boldsymbol{\varphi}^0$ and an initial $\boldsymbol{\mathcal{T}}^0$. Calculate the transformed level-set matrix at $t_{(j-1)}$; call this $\Psi(\boldsymbol{\mathcal{T}}^0, \boldsymbol{\psi})$.

For each component φ_i, propose a new value φ_i^*. Evaluate $\alpha(\varphi_i^*|\varphi_i)$, and apply the Reject/Accept criterion. If the proposal is accepted, update the component value: φ_i=φ_i^*; if the proposal is rejected, retain the component value φ_i= φ_i.

- Generate a new value $\mathcal{T}_h^*$, and calculate the corresponding transformed $\Psi(\boldsymbol{\mathcal{T}}_h^*, \boldsymbol{\psi})$.
- Evaluate $\alpha(\mathcal{T}_h^*|\mathcal{T}_h)$, and apply the Reject/Accept criterion.
- If the proposal is accepted then set $\mathcal{T}_h$=$\mathcal{T}_h^*$; otherwise $\mathcal{T}_h$=$\mathcal{T}_h$.

- Generate a new value $\mathcal{T}_r^*$, and calculate the corresponding transformed $\Psi(\boldsymbol{\mathcal{T}}_r^*, \boldsymbol{\psi})$.
- Evaluate $\alpha(\mathcal{T}_r^*|\mathcal{T}_r)$, and apply the Reject/Accept criterion.
- If the proposal is accepted then set $\mathcal{T}_r$=$\mathcal{T}_r^*$; otherwise $\mathcal{T}_r$=$\mathcal{T}_r$.

- Generate a new value $\mathcal{T}_v^*$, and calculate the corresponding transformed $\Psi(\boldsymbol{\mathcal{T}}_v^*, \boldsymbol{\psi})$.
- Evaluate $\alpha(\mathcal{T}_v^*|\mathcal{T}_v)$, and apply the Reject/Accept criterion.
- If the proposal is accepted then set $\mathcal{T}_v$=$\mathcal{T}_v^*$; otherwise $\mathcal{T}_v$=$\mathcal{T}_v$.

- Generate a new value $\mathcal{T}_z^*$, and calculate the corresponding transformed $\Psi(\boldsymbol{\mathcal{T}}_z^*, \boldsymbol{\psi})$.
- Evaluate $\alpha(\mathcal{T}_z^*|\mathcal{T}_z)$, and apply the Reject/Accept criterion.
- If the proposal is accepted then set $\mathcal{T}_z$=$\mathcal{T}_z^*$; otherwise $\mathcal{T}_z$=$\mathcal{T}_z$.

FIGURE 6.1: *The general structure of the estimation algorithm.*

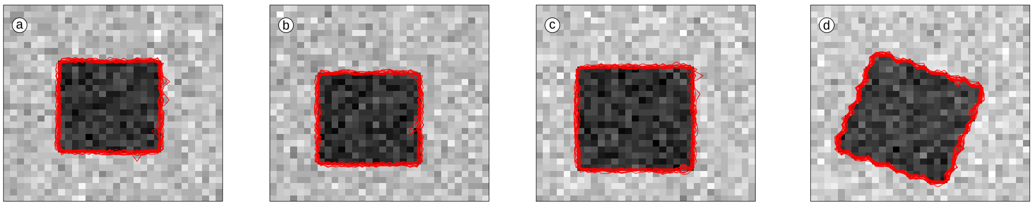


FIGURE 6.2: *Multiple boundary traces showing data images for a sequence of frames.*

For each frame, the MCMC algorithm was run for 20000 iterations, with a burn-in period of 2000 iterations. To remove the effects of autocorrelation, every 4th iteration is collected to produce a posterior sample of size 4500. In addition, the narrow-band technique can be applied where only components of the level-set matrix; near the boundaries are updated. The variances are adjusted at the beginning of the burn-in period to achieve an acceptance rate of about 23.4% (Roberts et al., 1997).

Experiment 1

In this example a simple dynamic simulation with a single moving object is considered. In particular, the sequence of images was simulated with a square object located around the centre, with the object moving and evolving with time. The data in the each frame has the intensities $\mu_* = 5$ and $\mu_0 = 7$ for the object and the background respectively, and a common variance $\sigma^2 = 0.3$. Then, to estimate the model parameters the proposed algorithm is used with $\beta = 0.6$, $\tau = 0.45$ and $\omega = 0.4$; there were chosen after initial experiments. Figure 6.2 shows the data image with a sub-sample of 20 curves chosen randomly from the sample of 4500 for all frames. It is clear that the variability is small at the boundaries which indicates the certainty in the location of the estimated boundary.

The posterior probability of the pixels is calculated showing what region, background or objects is most likely to occur in each pixel. The results for all frames are shown in Figure 6.3, where the posterior probabilities are close to 1 for all pixels which are known to be from the object region, and close to zero for those known to be from the background. Therefore, it can be said that the accuracy of

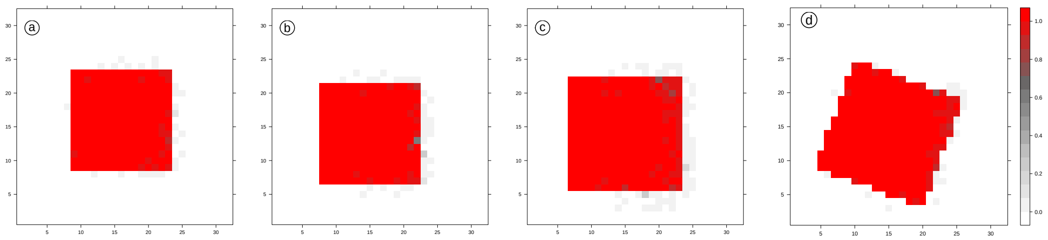


FIGURE 6.3: *The posterior probability of pixel i being part of the object for all frames.*

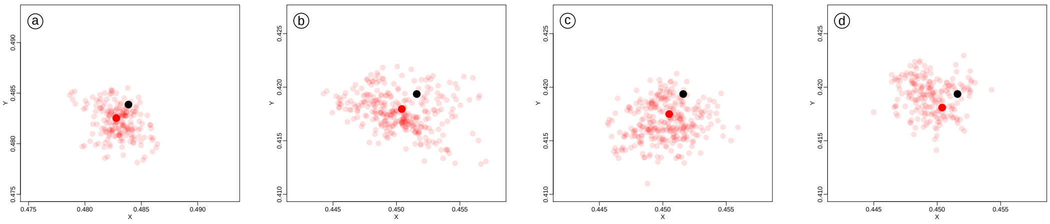


FIGURE 6.4: *A sub-sample of the location of the object centres for all frames, with the true location (in black).*

the object identification is very good.

Figure 6.4 shows a cloud of points for a sub-sample of the location of the object centres chosen randomly after the convergence of the algorithm, for all frames with the true location (in black). Clearly the variability is small and the true locations lie within this cloud of points which shows the certainty in the location of the object.

Parameter	True value	Post. mean (95 % CI)	Post. std dev	Post. skewness	Post. kurtosis
S_h	1.000	1.018 (0.628, 1.407)	0.238	-0.057	2.427
S_v	2.000	2.070 (1.667, 2.437)	0.243	-0.112	2.145
S_z	0.000	15.696 (-12.622, 42.605)	17.062	-0.139	2.981
S_r	0.000	0.007 (-0.086 0.093)	0.057	-0.126	2.394

TABLE 6.1: *Summary of posterior results for frame* 2.

First, the results obtained from the second frame, Figure 6.2(b), are discussed. Figure 6.5 shows the parameter traces (left column). The traces show rapid convergence and good random mixing. They also indicate moderate autocorrelation within the chain, with the autocorrelation functions reinforcing this (middle column). Also shown are estimates of the marginal posterior distribution (right column) as histograms and as smoothed density estimates. The thick line on the horizontal axis indicates the 95% credible intervals. Clearly, the small variation in these parameters shows that the parameters are well-estimated. In addition, all parameters have approximately symmetric distributions. The numerical summaries in Table 6.1 show that all of the posterior mean estimates are close to the true values, whilst the posterior standard deviations indicate that the parameters were estimated very precisely.

Parameter	True value	Post. mean (95 % CI)	Post. std dev	Post. skewness	Post. kurtosis
S_h	0.000	-0.154 (-0.835, 0.631)	0.441	0.280	2.424
S_v	0.000	-0.193 (-0.776, 0.600)	0.420	0.438	2.601
S_z	110	101.505 (5.733, 151.357)	45.073	-1.044	6.536
S_r	0.000	-0.002 (-0.026, 0.022)	0.015	0.093	3.114

TABLE 6.2: *Summary of posterior results for frame* 3.

Secondly, the results obtained from the third frame, Figure 6.2(c), are considered. Figure 6.6 shows the Markov chain paths (left column). The traces converge quickly for each parameter with good random fluctuations. The autocorrelation functions (middle column) show acceptable autocorrelations, although there is a slow decay in the ACF plot for parameter $\mathcal{T}_h$. Also shown are estimates of the marginal posterior distribution (right column) as histograms

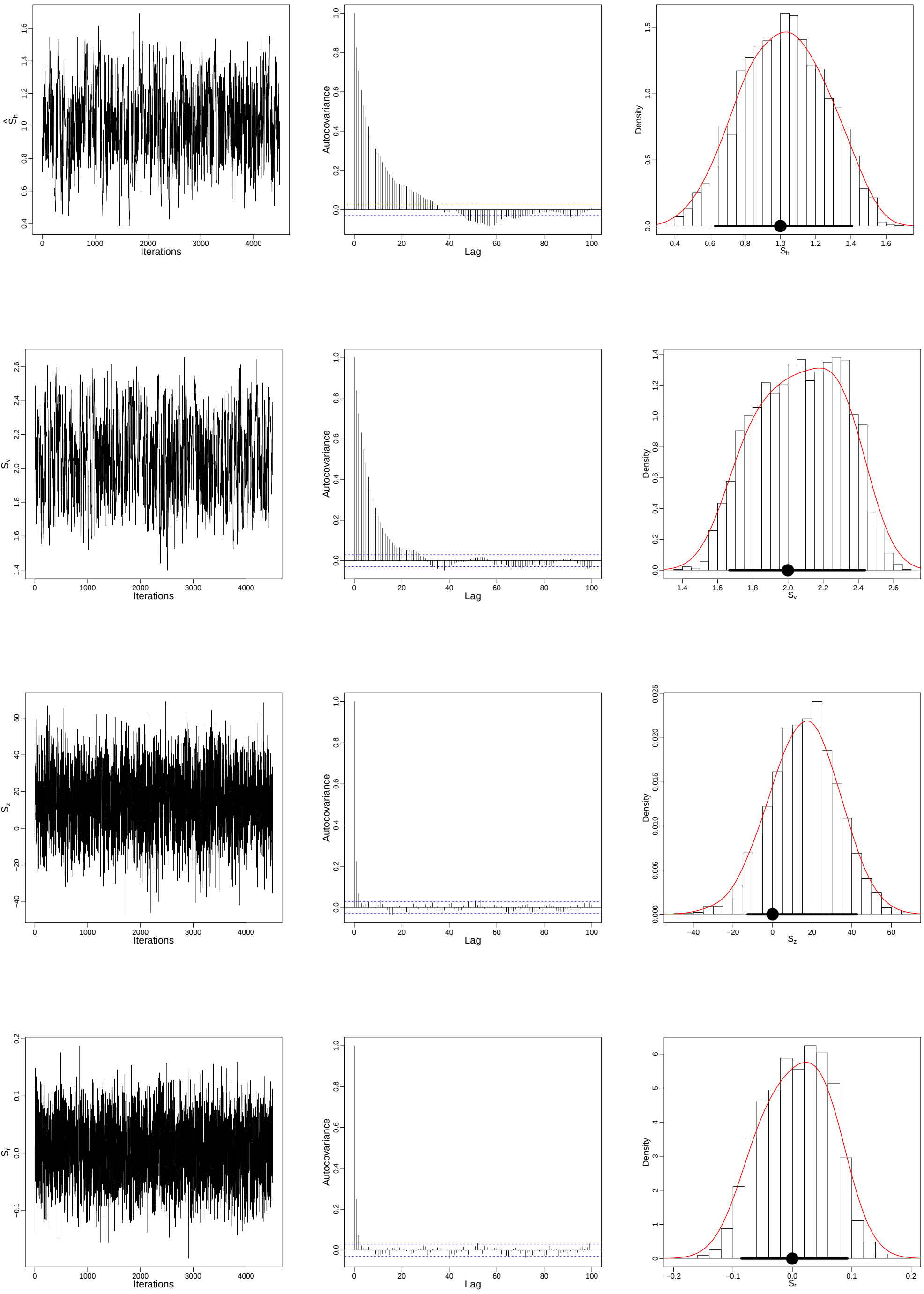


FIGURE 6.5: *Plot of MCMC traces (left column), autocorrelation function (ACF) (middle column), and histogram with density estimates (right column) for temporal parameters in frame* 2.

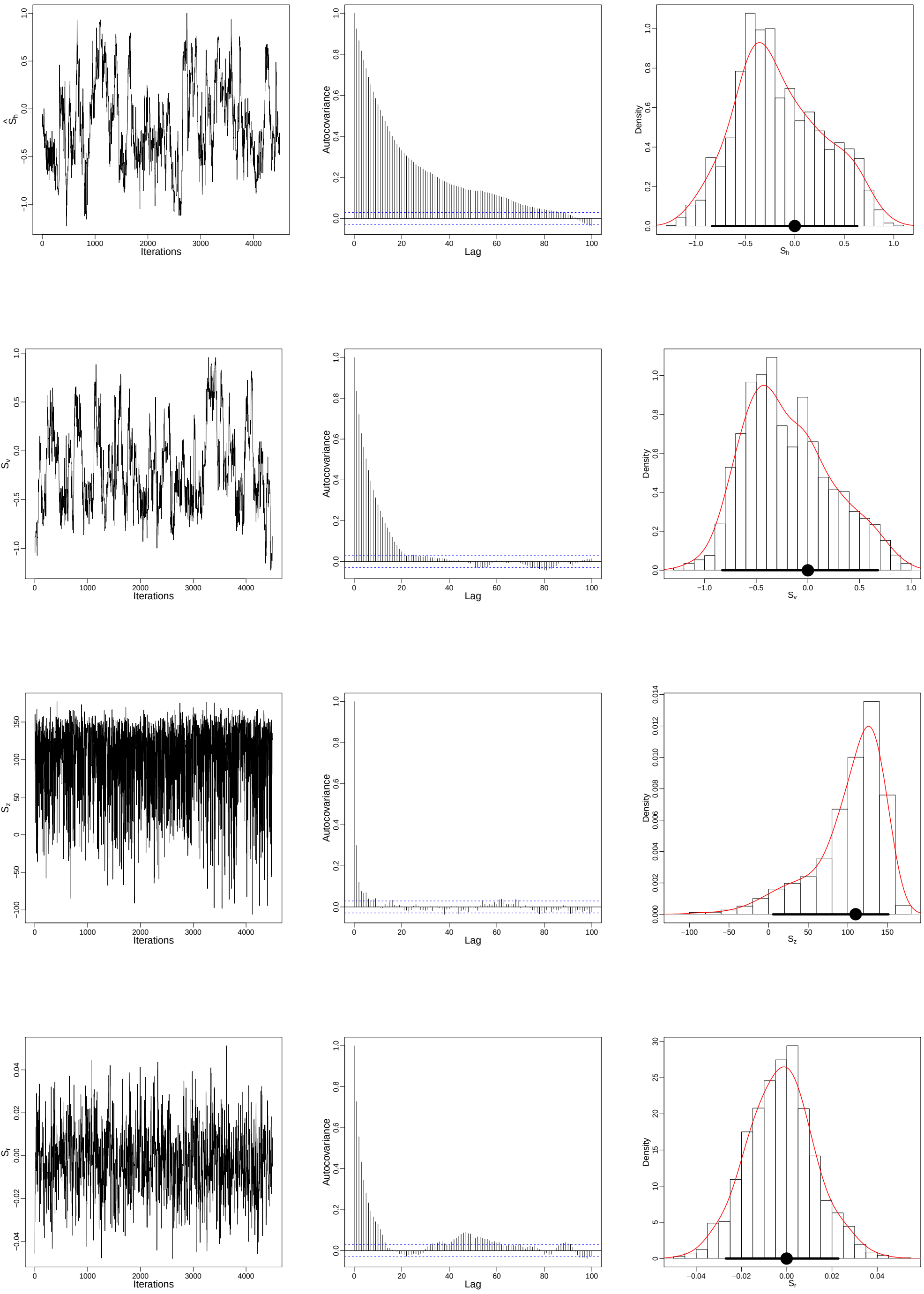


FIGURE 6.6: *MCMC traces (left column), autocorrelation function (middle column), and histograms and density estimates (right column) for temporal parameters in frame 3.*

and as smoothed density estimates. The thick line on the horizontal axis indicates the 95% credible intervals. The rotation angle has an approximately symmetric distribution. However, the sample distributions for $\mathcal{T}_h$, $\mathcal{T}_v$, and $\mathcal{T}_z$ are very asymmetric. The shift in the x and y-direction distributions are positively skewed, whilst the shift in the z-direction distribution is negatively skewed. The numerical summaries in Table 6.2 show that all of the posterior mean estimates are close to the true values, with the small values of the posterior standard deviations indicating that the parameters were estimated very precisely.

Finally, the results obtained from the last frame, Figure 6.2(d), are considered. Figure 6.7 shows the Markov chain paths (left column). The traces show rapid convergence and good random mixing. The autocorrelation functions (middle column) show acceptable autocorrelations, although again there is a slow decay in the ACF plot for parameter $\mathcal{T}_v$. Also shown are estimates of the marginal posterior distribution (right column) as histograms and as smoothed density estimates. The thick line on the horizontal axis indicates the 95% credible intervals. The shift in the x and y-directions have approximately symmetric distributions. In contrast, the shift in the z-direction and the rotation angle have negatively skewed distributions. The numerical summaries (Table 6.3) show of the posterior standard deviations indicate that the parameters were estimated very precisely.

Parameter	True value	Post. mean (95 % CI)	Post. std dev	Post. skewness	Post. kurtosis
S_h	0.000	-0.037 (-0.273 0.197)	0.141	0.075	3.103
S_v	0.000	0.048 (-0.163, 0.296)	0.137	0.411	3.430
S_z	0.000	45.150 (-7.687, 93.186)	24.431	-0.103	2.986
S_r	0.349	0.343 (0.299, 0.384)	0.025	-0.103	2.986

TABLE 6.3: *Summary of posterior results for frame* 4.

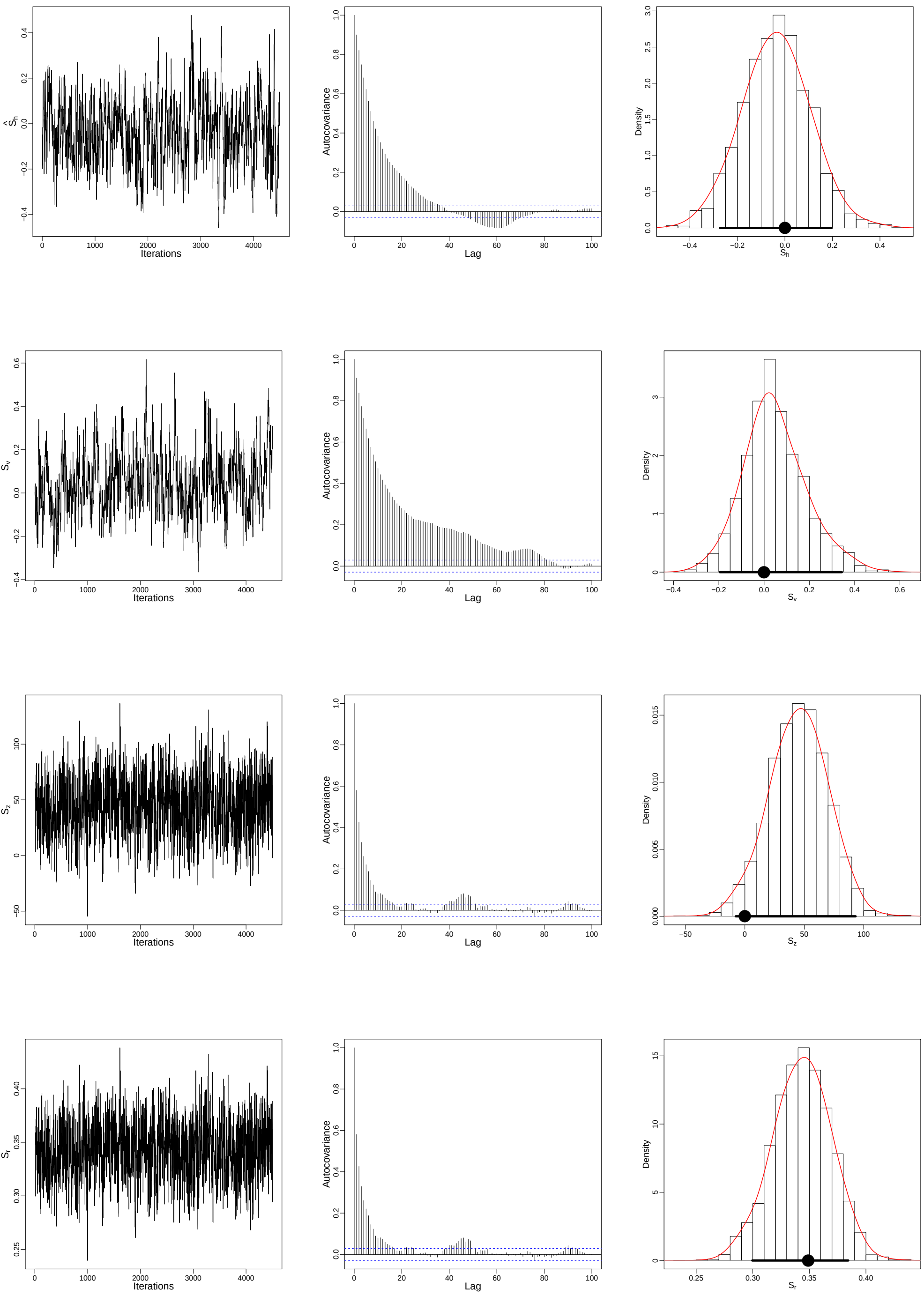


FIGURE 6.7: *MCMC traces (left column), autocorrelation function (middle column), and histograms and density estimates (right column) for temporal parameters in frame 4.*

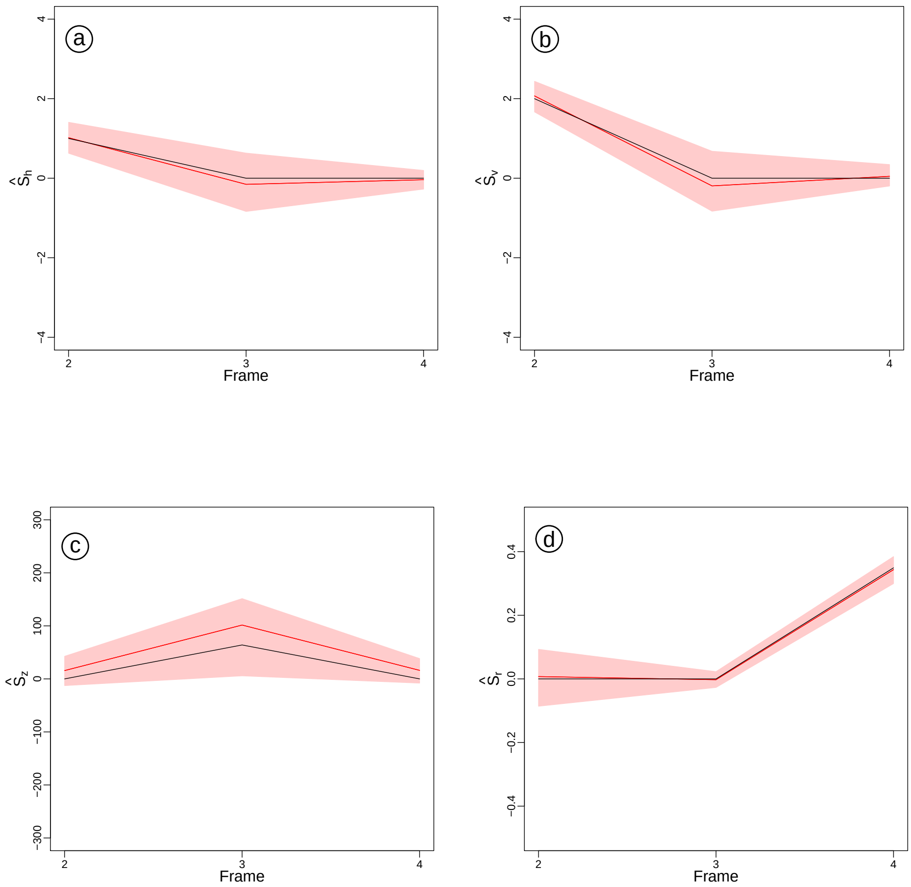


FIGURE 6.8: *Posterior estimates with* 95% *credible intervals and true values (in black): (a) shift in the x-direction,* $\mathcal{T}_h$*, (b) shift in the y-direction,* $\mathcal{T}_v$*, (c) shift in the z-direction,* $\mathcal{T}_z$*, and (d) rotation angle,* $\mathcal{T}_r$*.*

Figure 6.8 shows time series plots for the posterior estimates with the 95% credible intervals and true values (in black) for the shift in the x-direction parameter (a), shift in the y-direction parameter (b), shift in the z-direction parameter (c), and rotation angle (d). In each panel, the time sequence shows smooth changes with a generally constant credible interval width surrounding the estimated value. Notice that if zero is included in the credible interval for any of the temporal parameters then this indicates that there is no corresponding change in the object.

A rose diagram is a variation of a circular histogram, with the bars being replaced by sectors (Mardia and Jupp, 2009). The radii of each sector is proportional to the square root of the group frequency and so the area of each sector

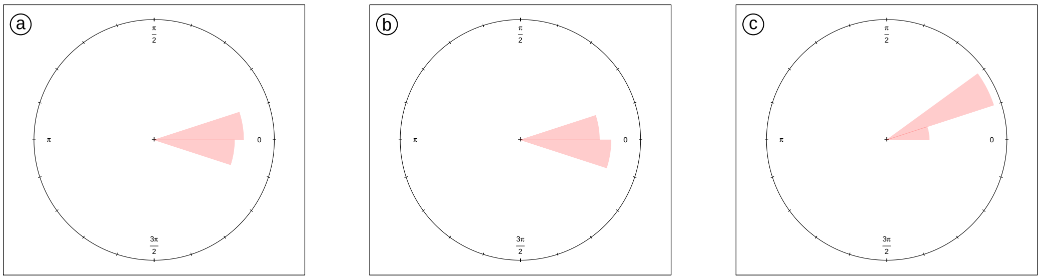


FIGURE 6.9: *Rose diagrams for rotation angles for frames: (a)* 2*, (b)* 3*, and (c)* 4*.*

is proportional to the group frequency. Figure 6.9 shows rose diagrams for the rotation angle parameter for all time. Each panel shows the rose diagram for the angle sample at a particular time. From the figure it is clear that there is a small variation in all samples of angle, which indicates that estimation of the rotation angle parameter is reliable.

	$\mathcal{T}_h$	$\mathcal{T}_v$	$\mathcal{T}_z$
$\mathcal{T}_v$	-0.014		
$\mathcal{T}_z$	0.005	0.006	
$\mathcal{T}_r$	-0.021	-0.006	0.003

(a)

	$\mathcal{T}_h$	$\mathcal{T}_v$	$\mathcal{T}_z$
$\mathcal{T}_v$	-0.015		
$\mathcal{T}_z$	-0.078	0.077	
$\mathcal{T}_r$	0.014	-0.051	-0.078

(b)

	$\mathcal{T}_h$	$\mathcal{T}_v$	$\mathcal{T}_z$
$\mathcal{T}_v$	-0.015		
$\mathcal{T}_z$	-0.078	0.077	
$\mathcal{T}_r$	0.014	-0.051	-0.078

(c)

TABLE 6.4: *Estimated correlations between parameters for frames: (a)* 2*, (b)* 3*, and (c)* 4*.*

The correlations between temporal parameters over time are calculated. The results are represented in Table 6.4 where each sub-table shows the parameter

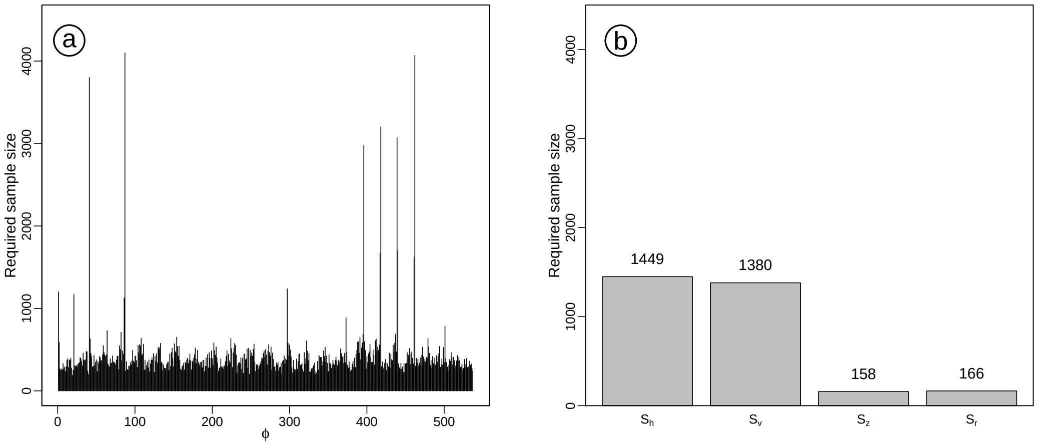


FIGURE 6.10: *Sample size calculation for the model parameters: (a) components of $\boldsymbol{\varphi}$ in the narrow-band region, and (b) temporal parameters.*

correlations for a specific time point. It is clear that there is no correlation between the different temporal parameters.
Sample size calculations were performed for each parameter (see Aykroyd and Green, 1991; Aykroyd and Mardia, 2003). Figure 6.10 show the result of the calculations for the components of the level-set matrix in the narrow-band region (a) and the temporal parameters (b); these results indicated that the main run of size 4500 is more than adequate.

Before moving on to the second experiment, the proposed method was applied using an image replaced by noise. This might happened if the camera stops recording at a specific time point, say for example at t_j. In this situation, the zero-level curve collapses and the algorithm breaks down after a few iterations. This is because the algorithm started with a big value of ω, hence there is no temporal information. This collapsing can be used as a warning alarm, and hence, in such cases the frame at t_j can be removed in the dynamic estimation. Then, the process continues to estimate the following frame at $t_{(j+1)}$ using the available data $\boldsymbol{y}^{t_{(j+1)}}$, and the estimated parameters from the previous frame at $t_{(j-1)}$. However, if the algorithm is modified, to use the solution from the previous time point as the initial zero level-set curve and if ω is chosen reasonably, then the zero level-set curve will not collapse. This is because there is prior information about the objects, such as their shapes, sizes and locations. This suggested modification is based on a single experiment and so more investigation is needed to confirm the outcome in a wider range of situations.

Experiment 2

In this example, a sequence of images were simulated with two circles moving with time in a smooth manner. At the beginning one object obscures the other, then both objects move in such a way that the covered object gradually becomes visible, at the end the objects are well separated. The data in each frame has the intensities $\mu_* = 5$ and $\mu_0 = 7$ for the objects and the background respectively, and a common variance $\sigma^2 = 0.6$. In the first frame, (a), the boundaries of the object were estimated individually because there is no temporal information. The proposed algorithm is then applied, with $\beta = 1$, $\tau = 0.45$ and $\omega = 0.7$, to track the objects, this is done by estimating the boundaries of the objects, the location of the objects, and the changes in the objects. To assess the convergence of the MCMC method, the plots of the Markov chain paths and the corresponding autocorrelation functions of the model parameters were checked. The plots show rapid convergence with acceptable autocorrelations.

Figure 6.11 shows the data with a sub-sample of 20 curves chosen randomly after the convergence of the algorithm. In each panel, it is clear that the variability is moderate at the boundaries which shows a reasonable certainty about the location of the estimated boundary. The posterior probability of the pixels is calculated showing what region, background or objects, is most likely to occur in each pixel. The results for all frames are shown in Figure 6.12 (right). The posterior probabilities are close to 1 for all pixels which are known to be from the object region, and close to zero for those known to be from the background. For all frames, it is quite clear that the posterior probabilities for most object pixels are roughly 1, with a few object pixels around the boundaries are less than 0.8. It can therefore be said that the objects has been tracked well throughout the sequence.

Figure 6.13 shows clouds of points for a sub-sample of the location of the object centres for frames 1 to 9 (a) and for frame 10 (b), with the true locations (in black). The locations are estimated for one object where the two circles are

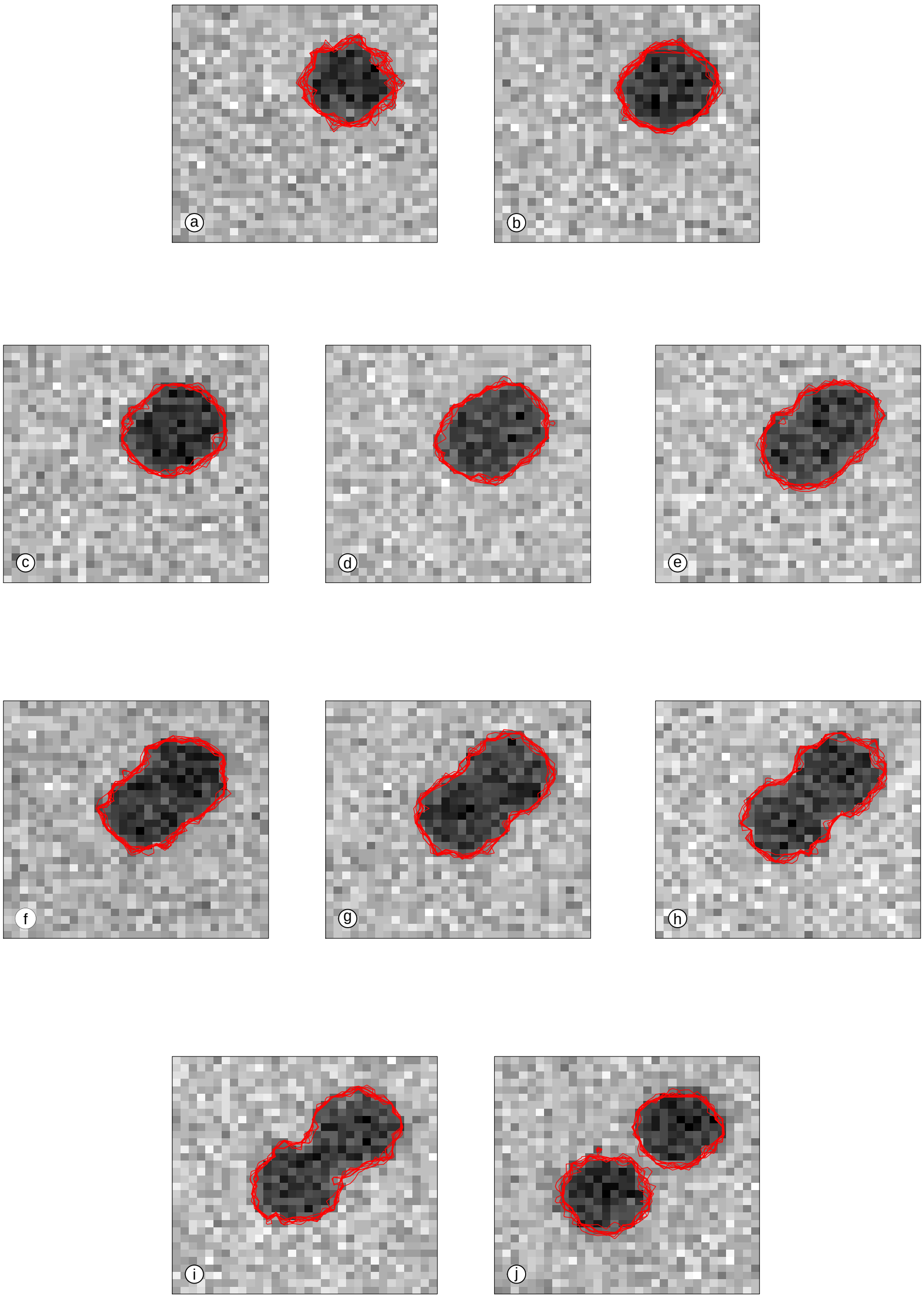


FIGURE 6.11: *Multiple boundary traces showing data images for a sequence of frames.*

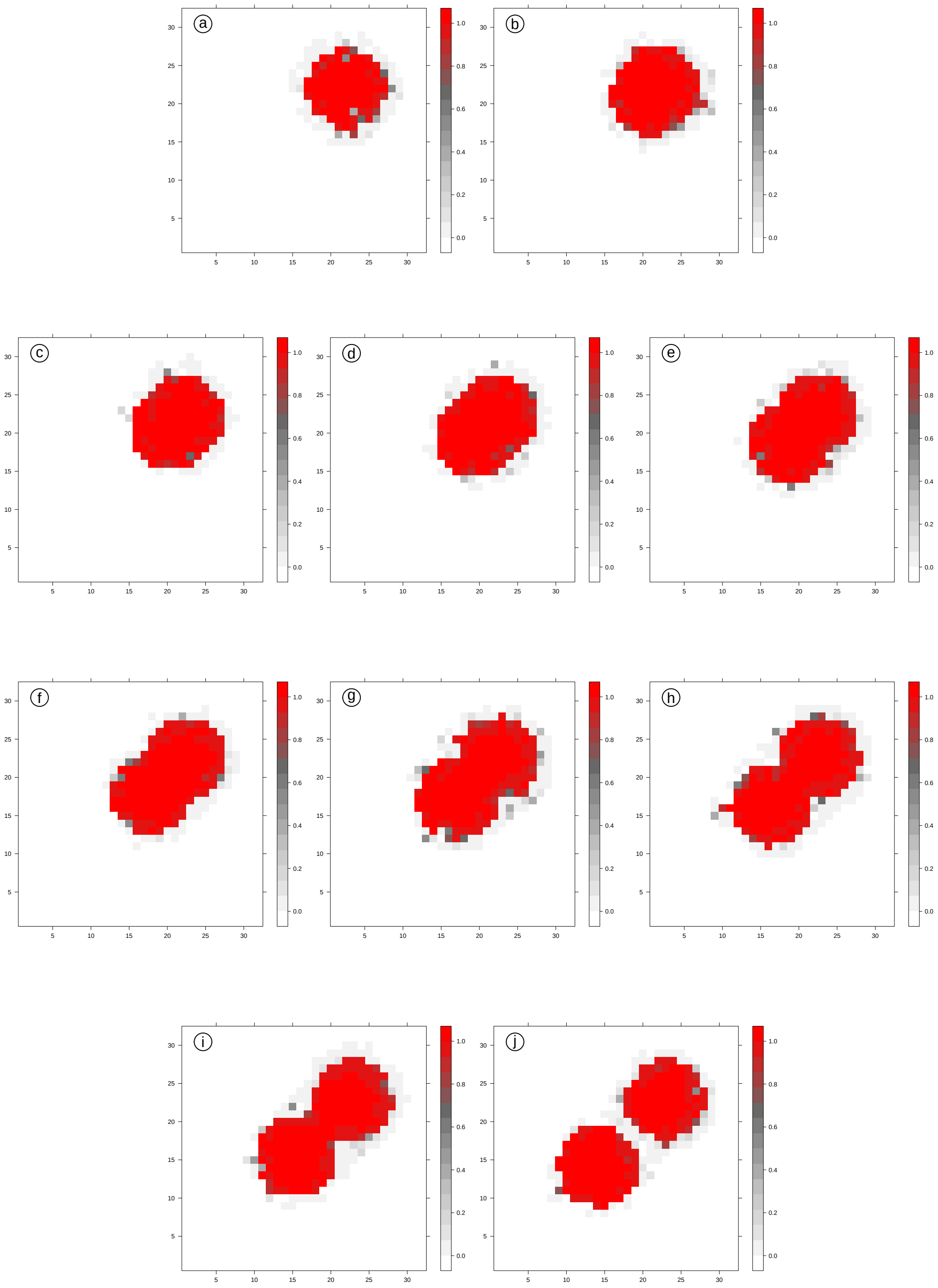


FIGURE 6.12: *The posterior probability of pixel i being part of the object for all frames.*

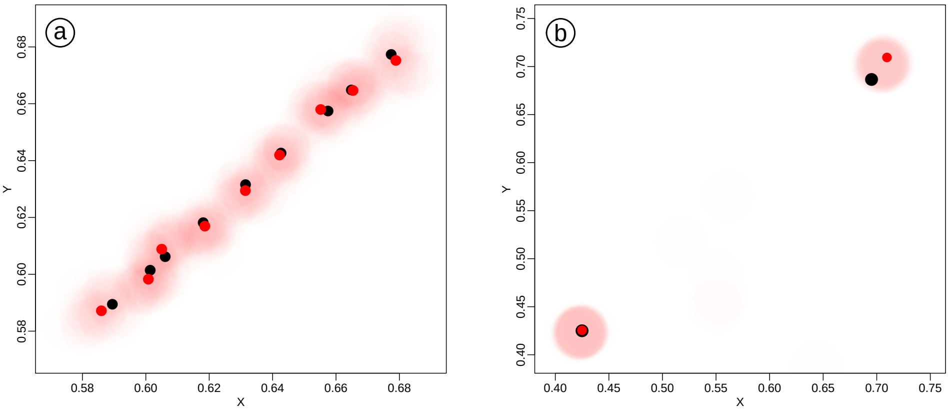


FIGURE 6.13: *Sub-samples of the locations of the object centres for frames* 1 *to* 9 *(a) and for the two objects in frame* 10 *(b), with the true locations (in black).*

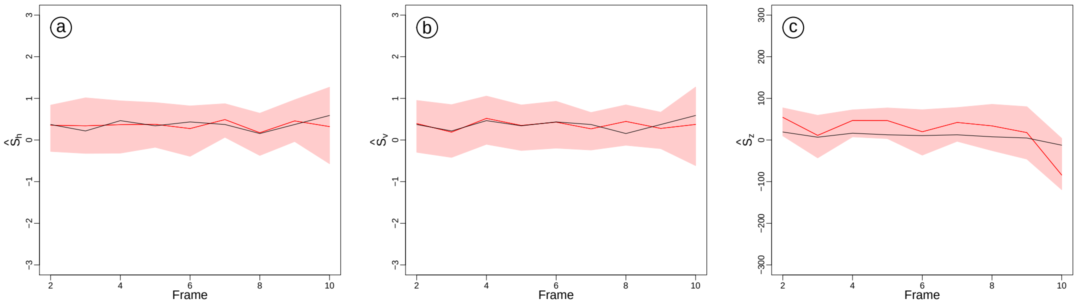


FIGURE 6.14: *Posterior estimates with the* 95% *credible intervals for the temporal parameters: (a)* $\hat{T}_h$, *(b)* $\hat{T}_v$, *and (c)* $\hat{T}_z$, *with true values (in black).*

overlapping in (a), and the location are estimated for the two circles after the separation in (b). For all frames, it is clear that the variation around the mean locations (in dark red) is small and the true locations (in black) lie within these clouds of points.

Figure 6.14 shows time series plots for the posterior estimates with the 95% credible intervals and true values (in black) for the shift in the x-direction parameter (a), shift in the y-direction parameter (b), and shift in the z-direction parameter (c). The figure shows the changes in locations and sizes of the objects were estimated very well. In each panel, the time sequence shows smooth changes with a generally constant credible interval width surrounding the estimated value. For the shift in the z-direction there is an abrupt change at the end of the sequence. This is because the two circles are well separated and so

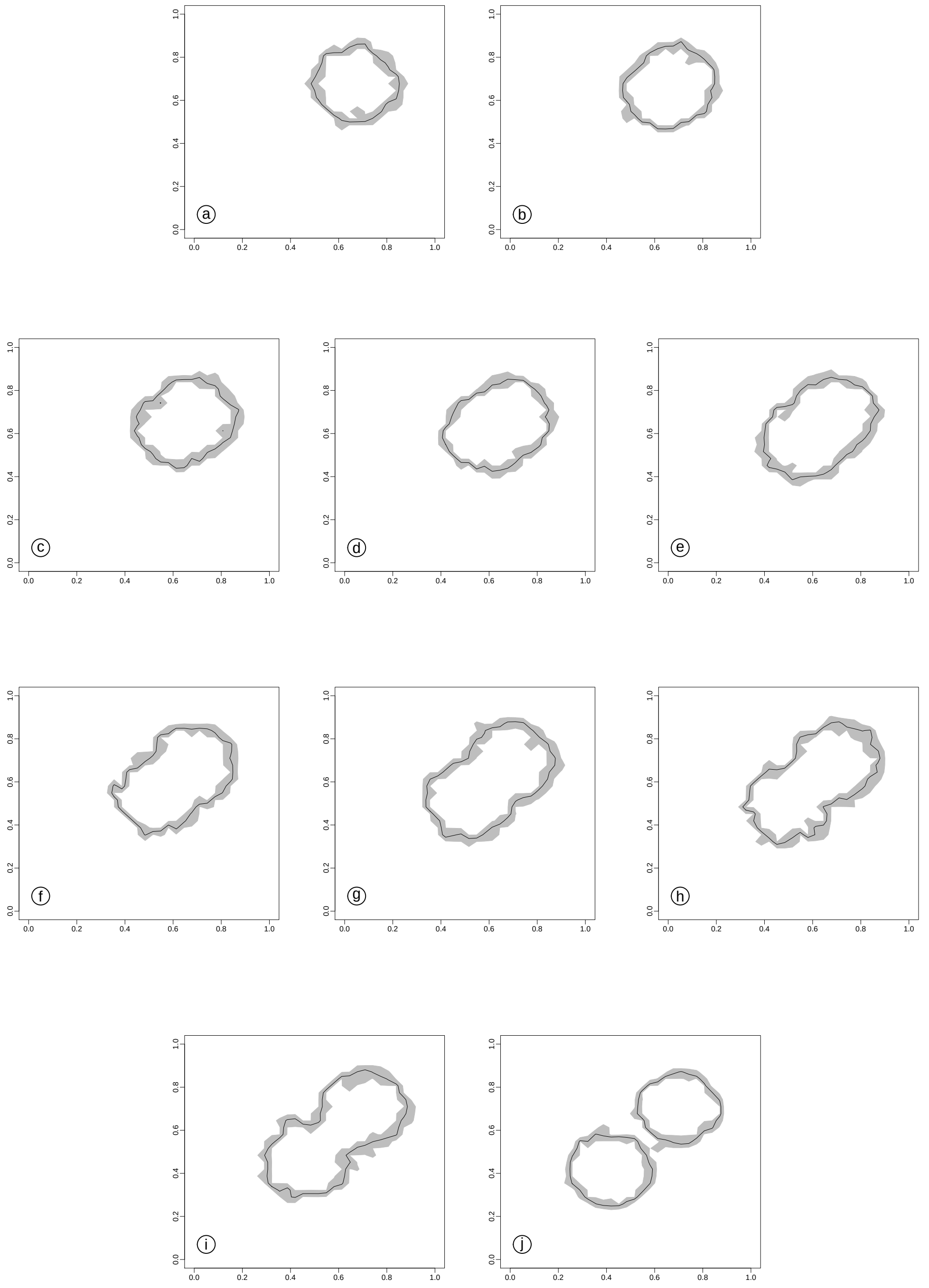


FIGURE 6.15: *The estimated boundaries for all frames (shown in black), with no temporal priors, with the* 95% *credible intervals for the object boundaries (shown in grey).*

the size of the objects reduces.

Figure 6.15 shows the credible intervals for the level-set curve for all frames, these results are obtained without using any temporal priors. Also, 6.16 shows

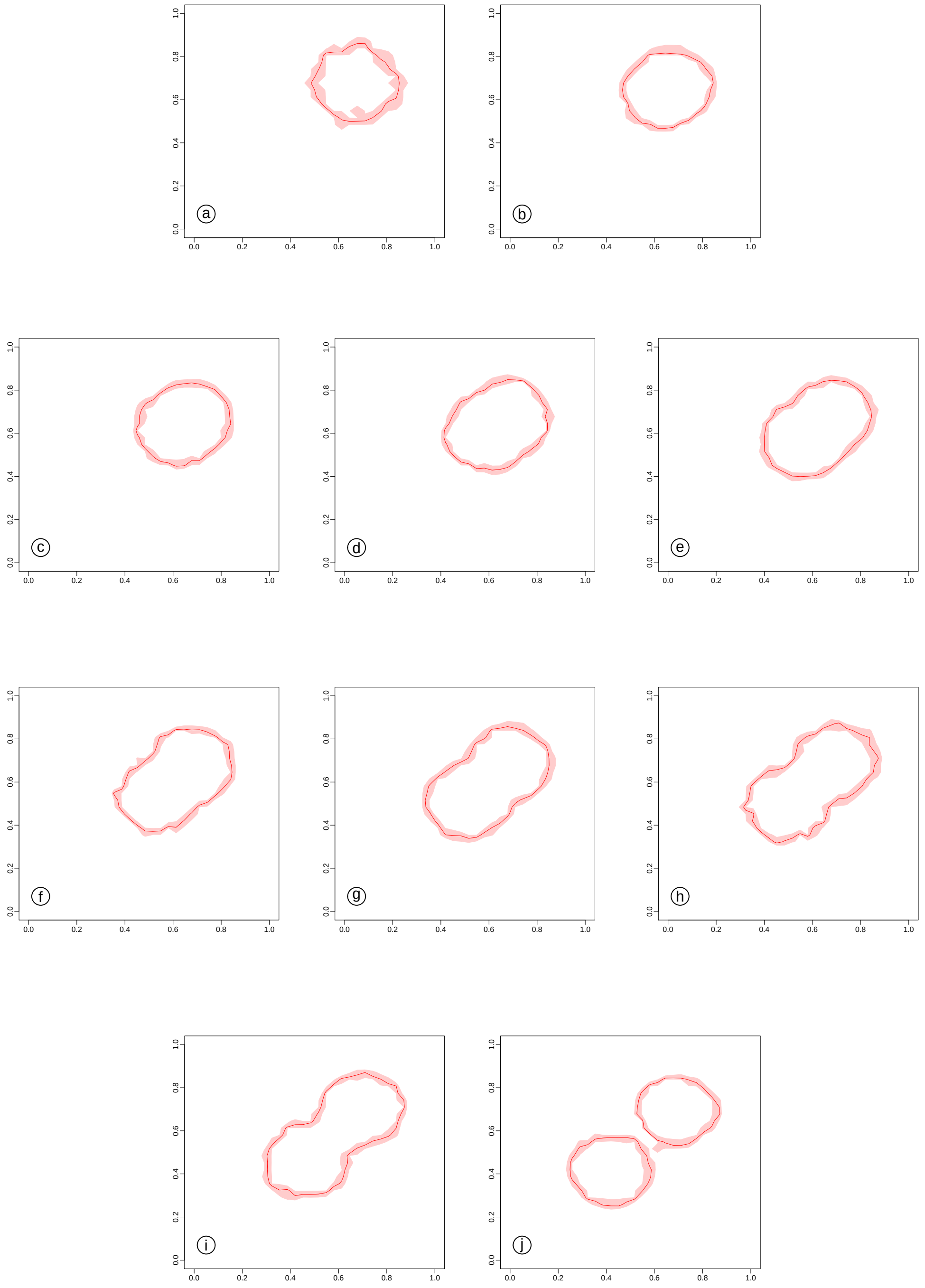


FIGURE 6.16: *The estimated boundaries for all frames (shown in red), with temporal priors, with the* 95% *credible intervals for the object boundaries (shown in pink).*

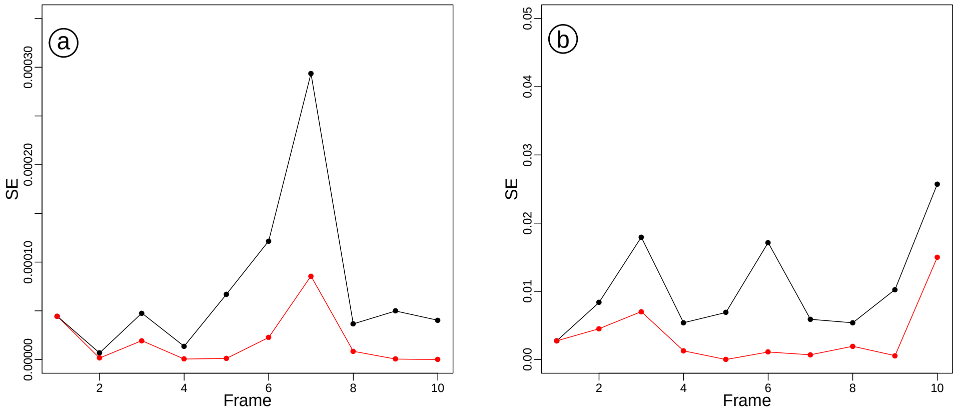


FIGURE 6.17: *The SE for the estimation of: (a) area of the objects and (b) length of the boundaries. With temporal prior (in red) and with no temporal prior (in black).*

the credible intervals for the level-set curve for all frames, these results are obtained with using temporal priors. In both figures, the regions of the credible intervals with no temporal priors (in grey) for all frames are compared to the regions of the credible intervals with temporal priors (in pink) for the corresponding frames. It is clear from this comparison, that the thickness of the pink regions in Figure 6.16 are smaller with much less variability around the estimated boundaries compared to the grey regions in Figure 6.15. This indicates that the inclusion of temporal priors has affected the parameter estimation, and that the variability in the estimated boundaries reduced.

In Figure 6.17, the SE of the estimated area of the objects (a) and length of the boundaries (b) for both models are compared for all frames. In each panel the red line represents the SE from the estimation with temporal priors, and the black line shows the SE from the estimation with no temporal priors. All SE values in both panels are very small, and so the estimation of area of the objects in both models is good. However, the estimation of the area of the objects with temporal priors gives a smaller SE for all frames compared to the estimation with no temporal priors. Similarly, the SE of the estimated length of the boundaries also shows a reduction for the estimation with temporal priors compared to the estimation with no temporal priors. This is true for all frames, even with a challenging case at the end where the two objects are separated.

6.5 Conclusions

This chapter has described a novel modelling approach through level sets for the use in dynamic imaging problems. Temporal information was easily incorporated into the estimation process using the Bayesian approach. A Monte Carlo sampler was used to explore the full posterior distribution, and to estimate many choices of the outputs, i.e. point estimates, credible intervals/regions, and marginal distributions for the parameters. A range of simulated images have been considered to illustrate the proposed algorithm. The results show that the method is able to estimate the temporal changes and to describe the relationships between temporal changes. In addition, it is clear that the inclusion of temporal prior information, makes a substantial improvement. For further investigation the method needs to be applied to a wider range of examples including real dynamic imaging situations. Also, a further extension could be considered to allow estimation errors to be propagated between frames due to the given level-set matrix from the previous frame also being estimated and hence involving uncertainty.

Chapter 7

Conclusions and further work

The aim of this thesis was to produce and investigate statistical methodologies for modelling image data through the level-set approach, and based on the experiments performed are successful in achieving this goal. In this final chapter, the main results presented in this thesis are summarized, and the future direction of the work is addressed.

7.1 Conclusions

The main contributions of this thesis are as follows:

- In Chapter 3 and Chapter 4, a rigorous Bayesian formulation of image segmentation based on the level-set method was introduced. This model was then generalized to allow the consideration of many important practical applications. In particular, a variety of prior models were considered as more flexible generalizations for prior information description. Initially, exponential distributions were used to model the area of objects and the length of the objects, before proposing beta and gamma distributions as more appropriate choices. Also, the Student's t-distribution is proposed as a more robust choice than a Gaussian distribution for modelling the errors. The results showed that performing a sensitivity analysis provides an important assessment of robustness of subjective model choices. The approach can easily be generalized, making it a feasible approach for more complex image problems.

- In Chapter 5, a new Bayesian modelling approach was proposed for object identification in image analysis through level sets. The MCMC estimation algorithm was also described for sampling from the posterior distribution. The proposed algorithm was applied to both simulated and real data. A range of outputs were summarized using the MCMC sampling procedure. The results clearly indicated that the combined Bayesian/MCMC procedure worked well.

- In Chapter 6, a new method for modelling the temporal evolution of images through level sets was produced; also, the MCMC algorithm was described for sampling the posterior distribution. The proposed algorithm was applied to simulated image sequences, and successfully estimated temporal components as well as other model parameters. The results indicate that the proposed method could become a useful tool for temporal modelling in dynamic imaging problems.

7.2 Further work

There are several suggestions that could be considered as extensions from this thesis.

- Consider other types of prior information with the aim of improving the estimation. However, if the prior parameters are not set correctly, the estimation results from the MCMC methods might be poor; thus, there is a need for developing the estimation algorithm, to allow all parameters to be well estimated. The suggestion for including these parameters in the estimation process is to fit a hierarchical Bayesian model, which will result in effective and flexible estimation algorithms. Other information can easily be incorporated into the estimation process, such as the smoothness of the boundaries.

- In recent years, Bayesian modelling approaches have provided an appropriate framework for controlling and describing the segmentation process, to deal with many inverse problems which can be considered as identifying objects. The proposed algorithms need to be developed, and then it will be appropriate to be applied in inverse imaging problems.

- Segmentation of overlapping objects, where parts of the objects are behind others is considered as one of the most challenging problems in image analysis. In the proposed algorithms, the number of objects can be estimated as well as other model parameters. However, this is extremely difficult for overlapping objects, as well as the identification of the boundaries, and hence these algorithms could be extended to consider such situations.

- In the temporal modelling approach proposed in this thesis, the given level-set function from the previous frame is, in fact, also estimated, and hence has attached uncertainty. If a sequence of estimates are calculated, then the accumulated uncertainty could be substantial. Hence, there is clearly scope to propose a method for the total uncertainty; this, of course, will lead to new insights into image sequence modelling.

References

Aykroyd, R. G. (2015). Statistical image reconstruction. In *Industrial Tomography: Systems and Applications*. pp. 401–427. Woodhead Publishing.

Aykroyd, R. G. (2018). A statistical approach to the inclusion of electrode contact impedance uncertainty in electrical tomography reconstruction. *International Journal of Tomography & Simulation*, **31**(1):56–67.

Aykroyd, R. G., Cattle, B. A., and West, R. M. (2016). Statistical estimation of interface-boundary shapes from electrical impedance tomographic measurements. *International Journal of Tomography & Simulation*, **29**(3):1–18.

Aykroyd, R. G. and Green, P. J. (1991). Global and local priors, and the location of lesions using gamma-camera imagery. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, **337**(1647):323–342.

Aykroyd, R. G. and Mardia, K. V. (2003). A wavelet approach to shape analysis for spinal curves. *Journal of Applied Statistics*, **30**(6):605–623.

Aykroyd, R. G. and Zimeras, S. (1999). Inhomogeneous prior models for image reconstruction. *Journal of the American Statistical Association*, **94**(447):934–946.

Bergeest, J.-P. and Rohr, K. (2012). Efficient globally optimal segmentation of cells in fluorescence microscopy images using level sets and convex energy functionals. *Medical Image Analysis*, **16**(7):1436–1444.

Berger, J. O., Moreno, E., Pericchi, L. R., Bayarri, M. J., Bernardo, J. M., Cano, J. A., De la Horra, J., Martín, J., Ríos-Insúa, D., Betrò, B., et al. (1994). An overview of robust Bayesian analysis. *Test*, **3**(1):5–124.

Besag, J. (1983). Discussion of paper by P. Switzer. *Bulletin of the International Statistical Institute*, **50**(3):422–425.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, **48**:259–302.

Besag, J. (1993). Towards Bayesian image analysis. *Journal of Applied Statistics*, **20**:107–119.

Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995). Bayesian computation and stochastic systems. *Statistical Science*, **10**(1):3–41.

Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society. Series B (Methodological)*, **55**(1):25–37.

Burger, M. and Osher, S. J. (2005). A survey on level set methods for inverse problems and optimal design. *European Journal of Applied Mathematics*, **16**(2):263–301.

Chan, T. F. and Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, **10**(2):266–277.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, **49**(4):327–335.

Coelho, L. P., Shariff, A., and Murphy, R. F. (2009). Nuclear segmentation in microscope cell images: a hand-segmented dataset and comparison of algorithms. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*. pp. 518–521. IEEE.

Cook, R. D. (1986). Assessment of local influence. *Journal of the Royal Statistical Society. Series B (Methodological)*, **48**(2):133–169.

Cremers, D., Rousson, M., and Deriche, R. (2007). A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape. *International Journal of Computer Vision*, **72**(2):195–215.

Cremers, D. and Soatto, S. (2003). Variational Space-time Motion Segmentation. In *Int. Conf. on Computer Vision*. pp. 886–893. IEEE.

Cremers, D. and Soatto, S. (2005). Motion Competition: A Variational Approach to Piecewise Parametric Motion Segmentation. *International Journal of Computer Vision*, **62**(3):249–265.

Davison, A. C. (2003). *Statistical Models*. Cambridge University Press.

Dryden, I. L., Farnoosh, R., and Taylor, C. C. (2006). Image segmentation using Voronoi polygons and MCMC, with application to muscle fibre images. *Journal of Applied Statistics*, **33**(6):609–622.

Dunlop, M. M., Iglesias, M. A., and Stuart, A. M. (2017). Hierarchical Bayesian level set inversion. *Statistics and Computing*, **27**(6):1555–1584.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics Springer, Berlin.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). *Bayesian Data Analysis*. Chapman & Hall/CRC Boca Raton, FL, USA.

Gelman, A., Roberts, G. O., and Gilks, W. R. (1996). Efficient Metropolis jumping rules. *Bayesian Statistics*, **5**(42):599–608.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(6):721.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science*, **7**(4):473–483.

Gharipour, A. and Liew, A. W.-C. (2016). Segmentation of cell nuclei in fluorescence microscopy images: An integrated framework using level set segmentation and touching-cell splitting. *Pattern Recognition*, **58**:1–11.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1995). *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC Interdisciplinary Statistics Series.

Green, P. J. (1990). Bayesian reconstructions from emission tomography data using a modified EM algorithm. *IEEE Transactions on Medical Imaging*, **9**(1):84–93.

Grenander, U. (1993). *General Pattern Theory*. Clarendon Press, Oxford.

Grenander, U. and MacRae Keenan, D. (1993). Towards automated image understanding. *Journal of Applied Statistics*, **20**(5-6):89–103.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**(1):97–109.

Hobolth, A., Pedersen, J., and Jensen, E. B. V. (2003). A continuous parametric shape model. *Annals of the Institute of Statistical Mathematics*, **55**(2):227–242.

Horritt, M. (1999). A statistical active contour model for SAR image segmentation. *Image and Vision Computing*, **17**(3-4):213–224.

Hurn, M., Steinsland, I., and Rue, H. (2001). Parameter estimation for a deformable template model. *Statistics and Computing*, **11**(4):337–346.

Iglesias, M. A., Lu, Y., and Stuart, A. M. (2015). A Bayesian level set method for geometric inverse problems. *arXiv preprint arXiv:1504.00313*.

Insua, D. R. and Ruggeri, F. (2012). *Robust Bayesian Analysis*. Springer Science & Business Media.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, **1**(4):321–331.

Little, R. J. and Rubin, D. B. (2014). *Statistical Analysis With Missing Data*. John Wiley & Sons.

Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media.

Mardia, K. (1996). The art and science of Bayesian object recognition. In *Int. Conf. on Image Fusion and Shape Variability Techniques, Leeds, UK, 3–5 July 1996*.

Mardia, K., Hainsworth, T., and Haddon, J. (1992). Deformable templates in image sequences. In *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*. pp. 132–135. IEEE.

Mardia, K. and Kanji, G. (1993). *Advances in Applied Statistics: Statistics and Images: 1*. Carfax Publishing Company.

Mardia, K. V. (1994). *Advances in Applied Statistics: Statistics and Images: 2*. Carfax Publishing Company.

Mardia, K. V. and Jupp, P. E. (2009). *Directional Statistics*. John Wiley & Sons.

McLachlan, G. and Peel, D. (2004). *Finite Mixture Models*. John Wiley & Sons.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, **21**(6):1087–1092.

Mitiche, A. and Aggarwal, J. K. (2016). *Computer Vision Analysis of Image Motion by Variational Methods*. Springer.

Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, **42**(5):577–685.

Nattkemper, T. W. (2004). *Automatic segmentation of digital micrographs: a survey. in Medinfo, pp. 847–851*.

Niforou, K. N., Anagnostopoulos, A. K., Vougas, K., Kittas, C., Gorgoulis, V. G., and Tsangaris, G. T. (2008). The proteome profile of the human osteosarcoma U2OS cell line. *Cancer Genomics-Proteomics*, **5**(1):63–77.

Osher, S. and Fedkiw, R. (2006). *Level Set Methods and Dynamic Implicit Surfaces*. Springer Science & Business Media.

Osher, S. and Fedkiw, R. P. (2001). Level set methods: an overview and some recent results. *Journal of Computational Physics*, **169**(2):463–502.

Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, **79**(1):12–49.

Park, H., Schoepflin, T., and Kim, Y. (2001). Active contour model with gradient directional information: directional snake. *IEEE Transactions on Circuits and Systems for Video Technology*, **11**(2):252–256.

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, **10**(4):339–348.

Phillips, D. and Smith, A. (1993). Dynamic image analysis using Bayesian shape and texture models. *Journal of Applied Statistics*, **20**(5-6):299–322.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Robert, C. and Casella, G. (2011). A short history of Markov chain Monte Carlo: subjective recollections from incomplete data. *Statistical Science*, **26**(1):102–115.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, **7**(1):110–120.

Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press.

Sokal, A. D. (1989). Monte Carlo methods in statistical mechanics: foundations and new algorithms. Technical report, Cours de troisieme cycle de la Physique en Suisse Romande, Lausanne.

Sussman, M., Smereka, P., and Osher, S. (1994). A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, **114**(1):146–159.

Tsai, R. and Osher, S. (2003). Review article: Level set methods and their applications in image science. *Communications in Mathematical Sciences*, **1**(4):1–20.

Uhlemann, S., Wilkinson, P. B., Chambers, J. E., Maurer, H., Merritt, A. J., Gunn, D. A., and Meldrum, P. I. (2015). Interpolation of landslide movements to improve the accuracy of 4d geoelectrical monitoring. *Journal of Applied Geophysics*, **121**:93–105.

Voss, J. (2013). *An introduction to statistical computing: a simulation-based approach*. John Wiley & Sons.

Wang, W., Ozolek, J. A., Slepcev, D., Lee, A. B., Chen, C., and Rohde, G. K. (2011). An optimal transportation approach for nuclear structure-based pathology. *IEEE Transactions on Medical Imaging*, **30**(3):621–631.

Weir, I. S. and Green, P. J. (1994). Modelling data from single-photon emission computerized tomography. *Journal of Applied Statistics*, **21**(1-2):313–337.

West, R. M., Aykroyd, R. G., Meng, S., and Williams, R. A. (2004). Markov chain Monte Carlo techniques and spatial–temporal modelling for medical EIT. *Physiological Measurement*, **25**(1):181–194.

West, R. M., Meng, S., Aykroyd, R. G., and Williams, R. A. (2005). Spatial-temporal modeling for electrical impedance imaging of a mixing process. *Review of Scientific Instruments*, **76**(7):073703.

Xie, J., Efendiev, Y., and Datta-Gupta, A. (2011). Uncertainty quantification in history matching of channelized reservoirs using Markov chain level set approaches. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Xing, F. and Yang, L. (2016). Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review. *IEEE Reviews in Biomedical Engineering*, **9**:234–263.

Zhang, W. and Li, H. (2017). Automated segmentation of overlapped nuclei using concave point detection and segment grouping. *Pattern Recognition*, **71**:349–360.