Image Segmentation using Markov Random Field Models

A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

Simon A. Barker (Wolfson College)

July 1998

Signal Processing and Communications Laboratory
Department of Engineering
University of Cambridge

Declaration

The research described in this dissertation was carried out by the author between October 1994 and July 1998. Except as indicated in the text, the contents are entirely original and are not the result of any work done in collaboration. No part of this dissertation has been submitted to any other university. This dissertation contains not more than 65000 words.

Simon A. Barker

Acknowledgements

I would like to thank my supervisor Dr. Peter Rayner for his invaluable advice and support over the last four years. I am also grateful to Dr. Bill Fitzgerald, Dr. Anil Kokaram and Dr. Simon Godsill for various enlightening discussions along the way and to Dr. Wu for many informative conversations on various aspects of Bayesian theory and of course Mahler. Thanks are also due to the various people who have strived to maintain the computer system. I would also like to thank my friends in the group, my friends from Surrey and Ayaz Majid for keeping me vaguely sane over the years. Finally, I am deeply indebted to my parents, who through their efforts and dedication instilled in me a respect for learning without which I would never have reached Cambridge.

Summary

The development of a fully unsupervised algorithm to achieve image segmentation is the central theme of this dissertation. Existing literature falls short of such a goal providing many algorithms capable of solving a subset of this highly challenging problem.

Unsupervised segmentation is the process of identifying and locating the constituent regions of an observed image, while having no prior knowledge of the number of regions. The problem can be formulated in a Bayesian framework and through the use of an assumed model unsupervised segmentation can be posed as a problem of optimisation. This is the approach pursued throughout this dissertation.

Throughout the literature, the commonly adopted model is an hierarchical image model whose underlying components are various forms of Markov Random Fields. Gaussian Markov Random Field models are used to model the textural content of the observed image's regions, while a Potts model provides a regularisation function for the segmentation.

The optimisation of such highly complicated models is a topic that has challenged researchers for several decades. The contribution of this thesis is the introduction of new techniques allowing unsupervised segmentation to be carried using a single optimisation process. It is hoped that these algorithms will facilitate the future study of hierarchical image models and in particular the discovery of further models capable of more closely fitting real world data.

The extensive literature surrounding Markov Random Field models and their optimisation is reviewed early in this dissertation, as is the literature concerning the selection of features to identify the textural content of an observed image. In the light of these reviews new algorithms are proposed that achieve a fusion between concepts originating in both these areas.

Algorithms previously applied in statistical mechanics form an important part of this work. The use of various Markov Chain Monte Carlo algorithms is prevalent and in particular, the reversible jump sampling algorithm is of great significance. It is the combination of several of these algorithms to form a single optimisation framework that lies at the heart of the most successful algorithms presented here.

Contents

Introduction 1

1.1 General Introduction

The segmentation of an observed image into an unknown number of distinct and in some way homogeneous regions remains a fundamental issue in low-level image analysis. There are many direct applications of such algorithms, for example: segmentation of ultrasound images; crop discrimination using images from synthetic aperture radar (SAR); segmentation of nuclear magnetic resonance (NMR) images; segmentation of X-ray images; remote sensing applications, and textile quality inspection. Alternatively, image segmentation can be viewed more generally as a critical early process providing input to the higher level processing schemes of a complex vision system. Many different methodologies have been applied to image segmentation, however a process that is both completely unsupervised and also robust has yet to be realised.

Unsupervised segmentation comprises the segmenting of an image into an *a priori* unknown number of regions. Over the previous two decades two types of algorithm have emerged as solutions to this problem: model fitting, reviewed in chapter 2, and more empirical, non-parametric classification algorithms, discussed in chapter 3.

The models reviewed in chapter 2 are hierarchical. Different spatial distributions are used to model each region of the image. The interaction between these regions is then modelled by a further spatial distribution which generally imposes a smoothing or regularisation constraint on possible segmentations. In the Bayesian sense, this second distribution forms a prior on the segmentation. Both of these models are usually non-causal and thus require extensive computation in their analysis. Throughout the 70's and early 80's algorithms [9] to optimise these models were in general *ad hoc* and were restricted to highly

supervised problems only. With computational power ever increasing, the development of the Gibbs Sampler [39] lead to an explosion of interest in the field. The ability to sample from complicated multivariate distributions using this and other MCMC techniques [77] has facilitated the analysis of highly generalised statistical models of which the hierarchical image model is perhaps one of the most challenging. However, even with these tools, the development of a single fully unsupervised segmentation process for the general hierarchical image model is a goal that has eluded researchers. To date, the only approach has been one of optimisation of a set of models and then model comparison to achieve a model selection. In effect this comprises an exhaustive search over the model set, a highly inefficient and unsophisticated methodology. Advancement beyond such algorithms has been facilitated by the development of the reversible jump Markov Chain Monte Carlo (MCMC) sampling algorithm [43]. This has proved the necessary catalyst to researchers working in a wide range of model selection problems. Hence, the use of this algorithm together with a variety of MCMC sampling techniques, specifically composition sampling, the Swendsen-Wang algorithm [90] and the Partial Decoupling algorithm [50] form the basis of the unsupervised segmentation algorithms developed throughout this dissertation.

The second extensive contribution towards image segmentation literature comprises the body of work in which the classification of regions is achieved using observed features. To calculate the necessary features the use of windows is required, thus producing segmentations at a coarser scale than that of the original image. This drawback is usually circumvented by using these window based unsupervised methods to achieve a rough segmentation from which parameters pertaining to a hierarchical image model (of the type reviewed in chapter 2) may be estimated. The model, estimated parameters and coarse segmentation can then be used to achieve a fine-scale segmentation using a supervised model-based segmentation algorithm.

There are several drawbacks to such a solution: the use of windowing in the coarse segmentation process will in general average the contribution of small features, thus they are unlikely to be represented in the eventual segmentation; there is an implied trade-off between the necessary size of windows needed to differentiate between textures and the accuracy of localisation of boundaries; the statistics or features are often *ad hoc*; finally, to achieve the coarse unsupervised segmentation, clustering processes are used which require the *a priori* setting of thresholds. However these algorithms have one major advantage over their model selection counterparts, their speed of convergence is far superior in terms of computation.

Given these two areas of prior research, the philosophy behind the algorithms developed throughout this dissertation is to utilise the advantages of the hierarchical image model, specifically the Bayesian framework, but also to formulate a single optimisation process capable of robust unsupervised segmentation. In chapter 4 this is achieved through the use of reversible jumps and composition sampling, however the optimisation is highly computationally intensive and for this reason, the use of very complicated models for individual regions is intractable.

To improve upon these procedures, more complex sampling procedures are implemented in chapter 5. Auxiliary variable techniques are used to both speed up the convergence of the segmentation process and to enhance the acceptance rate of the reversible jump sampling process used in the optimisation over differing image models. The technique used, specifically partial decoupling also allows the incorporation of the non-parametric features reviewed in chapter 3 into the hierarchical Bayesian framework to enhance convergence while leaving the eventual segmentation unaffected.

The incorporation of non-parametric statistics into a Bayesian framework through the use of hyper-parameters for auxiliary variables is an extremely powerful tool and as such need not be specific to the problem of image segmentation. As stated in the concluding chapter 6, the application of the techniques of chapter 5 to the less complicated problem of segmenting non-causal time series would be relatively straightforward.

1.2 Thesis Overview

Chapter 2 provides a review of existing model based image segmentation techniques. The criteria for supervised, semi-unsupervised and fully unsupervised segmentation algorithms are defined. The hierarchical Markov Random Field (MRF) image model is introduced with sections describing features of two widely used models, the Potts model and the Gaussian MRF. The use of line processes to model image boundaries is also examined. Segmentation techniques are then described beginning with those used to solve the supervised segmentation problem. Multi-resolution techniques are then discussed, including elements from renormalisation group theory. Next, semi-unsupervised segmentation algorithms are described, beginning with Maximum *a posteriori* (MAP) approaches, then continuing with the expectation-maximisation (EM) algorithm. The theory of mean fields follows with a

description of mean-field annealing in comparison to soft-decision EM techniques. Unsupervised segmentation methodologies and criteria are addressed before concluding with some observations regarding the relevance of the reviewed literature to the direction of research presented in this dissertation.

Chapter 3 describes the second large body of research concerning unsupervised segmentation. The approach examined here comprises classification of an image into regions through algorithms in a particular feature space. Three major types of feature are described together with the relevant classification or segmentation techniques. First introduced are model based statistics, then local spatial statistics such as residuals or those derived from cooccurrence matrices follow. Finally the multi-channel filtering approach to texture segmentation is examined and in particular, the application of the Gabor Wavelet function.

Chapter 4 presents new unsupervised segmentation algorithms for both an hierarchical Isotropic MRF model and a GMRF model are introduced. An important element of these algorithms is the reversible jump sampler, hence this is first introduced. The models are defined before the description of their associated optimisation algorithms and the presentation of some experimental results.

Chapter 5 describes further algorithms which comprise improvements in terms of convergence rate and computation requirements to those of Chapter 4. The algorithms utilise an auxiliary variable technique known as partial decoupling. This, together with its predecessor, the Swendsen-Wang algorithm is first outlined before the complete algorithms are described. A further algorithm used to achieve unsupervised segmentation for a hierarchical model including a line process is then given. Results are presented for all algorithms before conclusions are drawn.

Chapter 6 concludes the thesis with suggestions for further work.

Model Based Segmentation 2

2.1 Introduction

Segmentation is the process of splitting an observed image into its homogeneous or constituent regions. Segmentation may also be thought of as a labelling process, where each pixel in the observed image is assigned a label designating the region or class to which it belongs.

This formulation of the segmentation problem leads naturally to a hierarchical model. If the observed image $\mathbf{y}$ is defined on a rectangular $M \times N$ lattice $\mathbf{\Omega}$, indexed by the pair (i, j) so that $\mathbf{\Omega} = \{(i, j); 1 \leq i \leq M,\ 1 \leq j \leq N\}$, then the labels $\mathbf{x}$ may also be defined on an identical lattice. Thus, for each site $s = (s_i, s_j)$ there is a label x_s specifying to which region the observed pixel y_s belongs. The relationship between the observed gray-scale values and the labels is given by a set of Bayesian Likelihood functions. Thus if a single homogeneous region is labelled by c, it will comprise the set $\mathbf{c} : x_s = c$, and its likelihood will be modelled by the probability distribution $p\big(\{\mathbf{y}_s, s : x_s = c\} \mid \boldsymbol{\theta}_c^{(Y)}\big)$, where $\boldsymbol{\theta}_c^{(Y)}$ are the model parameters.

A further distribution acts as a prior on the labels $p(\mathbf{x} \mid \boldsymbol{\theta}^{(X)})$; here $\boldsymbol{\theta}^{(X)}$ denotes the associated model parameter vector. Thus the segmentation problem may be expressed as an optimisation problem over the labels of the entire image:

$$\hat{\mathbf{x}}_{MAP} \;=\; \arg\max_{\mathbf{x} \in \mathbf{X}} \; p(\mathbf{x} \mid \boldsymbol{\theta}^{(X)}) \prod_{c \in \mathbf{\Lambda}} p\big(\{\mathbf{y}_s, s : x_s = c\} \mid \boldsymbol{\theta}_c^{(Y)}\big) \tag{2.1}$$

where $\mathbf{\Lambda}$ is the set of possible region labels or classes and $\mathbf{X}$ is the set of all possible configurations of labels over the lattice, $\mathbf{\Omega}$. Thus with reference to the concept of a hierarchical model, the higher level label elements x_s index the likelihood function used as the image

model at each individual pixel. The attribution of these labels to sites via equation 2.1 defines the *supervised* segmentation problem. The problem is supervised in the sense that both the number of regions and the model parameters are known *a priori*. Procedures to solve this problem will be discussed further in section 2.3.

The above equation gives the Maximum *a posteriori* (MAP) estimate for the site labels. Another, less widely used criterion exists, the maximiser of posterior marginals (MPM) [70]. This criterion effectively minimises the expected number of mis-classifications per pixel. The MPM criterion is specified for a single pixel site by

$$\begin{aligned} \hat{x}_s &= \arg\max_{x_s \in \mathbf{\Lambda}} \; p(x_s \mid \mathbf{y}, \boldsymbol{\theta}^{(X)}) \\ &= \arg\max_{x_s \in \mathbf{\Lambda}} \sum_{\mathbf{x}_{\{\mathbf{\Omega}\setminus s\}} \in \mathbf{X}_{\{\mathbf{\Omega}\setminus s\}}} p\big(\mathbf{x} \mid \boldsymbol{\theta}^{(X)}\big) \prod_{c \in \Lambda} p(\{\mathbf{y}_s, s : x_s = c\} \mid \boldsymbol{\theta}_c^{(Y)}) \end{aligned} \tag{2.2}$$

where $\mathbf{x}_{\{\mathbf{\Omega}\setminus s\}}$ denotes the labels at all sites excepting site s. The difficulty of evaluating the summation has precluded the MPM's widespread applicability to the more complex texture models currently in vogue.

To extend the segmentation problem further, consider the possibility that the model parameters $\boldsymbol{\theta} \in \mathbf{\Theta}$, where $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(X)},\ \boldsymbol{\theta}_c^{(Y)},\ c \in \mathbf{\Lambda}\}$ may also be unknown, then the segmentation problem is described throughout this dissertation as *semi-unsupervised*. The criteria for segmentation now takes one of two forms,

$$\hat{\mathbf{x}}_{MAP},\ \hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\mathbf{x} \in \mathbf{X}, \boldsymbol{\theta} \in \mathbf{\Theta}} \; p(\mathbf{x} \mid \boldsymbol{\theta}^{(X)}) \prod_{c \in \mathbf{\Lambda}} p\big(\{\mathbf{y}_s, s : x_s = c\} \mid \boldsymbol{\theta}_c^{(Y)}\big) \tag{2.3}$$

$$\hat{\mathbf{x}}_{MAP},\ \hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\mathbf{x} \in \mathbf{X}, \boldsymbol{\theta} \in \mathbf{\Theta}} \; p(\boldsymbol{\theta})\, p(\mathbf{x} \mid \boldsymbol{\theta}^{(X)}) \prod_{c \in \mathbf{\Lambda}} p\big(\{\mathbf{y}_s, s : x_s = c\} \mid \boldsymbol{\theta}_c^{(Y)}\big) \tag{2.4}$$

the difference being between the selection of the maximum likelihood (ML) or the MAP criteria for model parameter estimation. Various algorithms have been proposed to evaluate these criteria and thus find estimates for the segmentation and model parameters. These will be described in section 2.5, but a fully robust algorithm has yet to be developed.

The final extension of the segmentation problem is known as *unsupervised segmentation*. Here the number of labels or regions is not known prior to segmentation. The problem is one of model order selection, where the model order $k \in \mathbf{K}$ gives the number of regions.

Thus far, one approach has been adopted,

$$\hat{\mathbf{x}}_{MAP},\ \hat{\boldsymbol{\theta}}_{ML},\ \hat{k} \ = \ \arg \max_{\mathbf{x}\in\mathbf{X}_{(\mathbf{k})},\ \boldsymbol{\theta}_{(k)}\in\boldsymbol{\Theta}_{(k)},\ k\in\mathbf{K}} \ p(\mathbf{x} \mid \boldsymbol{\theta}_{(k)}^{(X)}) \prod_{c\in\boldsymbol{\Lambda}_{(k)}} p\big(\{\mathbf{y}_s, s : x_s = c\} \mid \boldsymbol{\theta}_{(k)_c}^{(Y)}\big) + \mathrm{IC}(k, \boldsymbol{\theta}_{(k)}) \tag{2.5}$$

where $\mathrm{IC}(k, \boldsymbol{\theta}_{(k)})$ represents some form of information criterion. Existing algorithms which attempt to evaluate this optimisation are discussed later, in section 2.6, but for comparison here, the criterion for which the algorithms of chapters 4 and 5 of this thesis are introduced, will be specified here. Specifically, the MAP criteria is used to estimate all parameters;

$$\hat{\mathbf{x}}_{MAP},\ \hat{\boldsymbol{\theta}}_{MAP},\ \hat{k}_{MAP} \ = \ \arg \max_{\mathbf{x}\in\mathbf{X}_{(\mathbf{k})},\ \boldsymbol{\theta}_{(k)}\in\boldsymbol{\Theta}_{(k)},\ k\in\mathbf{K}} \ p(k)\ p(\boldsymbol{\theta}_{(k)})\ p(\mathbf{x} \mid \boldsymbol{\theta}_{(k)}^{(X)}) \times \prod_{c\in\boldsymbol{\Lambda}_{(k)}} p\big(\{\mathbf{y}_s, s : x_s = c\} \mid \boldsymbol{\theta}_{(k)_c}^{(Y)}\big) \tag{2.6}$$

The structures that constitute the underlying elements of the hierarchical model (both used throughout this thesis and throughout much of the relevant literature) comprise various types of Markov Random Fields (MRF's). Their widespread use has followed the seminal papers by Besag [11] and Geman & Geman [39], both of which provided new methods by which such complex models could be optimised to fit observed data. The theme of this dissertation follows a similar trend to their work, presenting new algorithms to fit hierarchical MRF models to data. By way of introduction, the remainder of this chapter gives a review of both the types of hierarchical models used in image segmentation and the methodologies by which they have previously been optimised.

2.2 Models

The MRF's used throughout the literature will be elaborated upon throughout this section but some inherent properties and definitions must first be stated (more complete reviews of fundamental MRF material can be found in [73] [67] [9] [39]).

The Markov Random Field forms a probabilistic model for a set of variables that interact on a lattice structure. The distribution for a single variable at a particular site

is conditioned on the configuration of a predefined neighbourhood surrounding that site. This effectively defines the Markov property of the process: the process is Markov not in the causal or even the bilateral sense, but with respect to this particular neighbourhood structure.

To link the MRF's structure to a joint probability distribution Hammersley & Clifford [9] posed the question,"*given the neighbours at each site, what is the most general form which* $U(\mathbf{x})$ *may take in order to give a valid probability structure to the system?*" Here, $U(\mathbf{x})$ is the energy function or Hamiltonian of a Gibbs distribution;

$$p(\mathbf{x}) \;=\; \frac{1}{Z}\exp\Big\{-\frac{1}{T}U(\mathbf{x})\Big\} \tag{2.7}$$

where, T is a constant analogous to temperature and Z is the normalising constant or *partition function* of the system. Hammersley & Clifford's famous theorem [9] (curiously unpublished by the original authors) states that "*given the neighbourhood structure of the model, for any set of sites within the lattice, their associated contribution to the Gibbs energy function should be non-zero, if and only if the sites form a clique.*[1] *The contribution of each clique may be assigned arbitrarily.*"

To prove this theorem the effect of altering the state at a single site was examined on the joint energy function for the entire MRF. Thus a comparison was made between probability distributions for non-zero values of the variable x_i and the case when $x_i = 0$. Letting $\mathbf{x}$ and $\mathbf{x}'$ denote the ensemble of states in these two realisations, their ratio can be written

$$\exp\{-U(\mathbf{x}) + U(\mathbf{x}')\} \;=\; \frac{p(x_i \mid x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)}{p(x_i = 0 \mid x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)} \tag{2.8}$$

Expansion of $U(\mathbf{x})$ into a general polynomial,

$$\begin{aligned} U(\mathbf{x}) \quad = \quad & \sum_{1\le i\le n} x_i G_i(x_i) \;+\; \sum_{1\le i\le n-1}\;\sum_{i<j\le n} x_i x_j G_{i,j}(x_i, x_j) \\ & + \sum_{1\le i\le n-2}\;\sum_{i<j\le n-1}\;\sum_{j<k\le n} x_i x_j x_k G_{i,j,k}(x_i, x_j, x_k) \\ & + \;...\; + \; x_1 x_2 ... x_n G_{1,2,...,n}(x_1, x_2, ..., x_n) \end{aligned} \tag{2.9}$$

where the $G(\cdot)$'s are unspecified functions, allows the difference in energies of equation 2.8 to be written explicitly. For example consider the case when i=1, the difference in energies

[1] A *clique* is defined as a set of lattice sites that are all neighbours of each other. See [67] or [73] for a more detailed definition.

between the case when $x_1 = 0$ and $x_1 \neq 0$ will be given by

$$\begin{aligned} U(\mathbf{x}) - U(\mathbf{x}') \quad = \quad & x_1 \Big\{ G_1(x_1) \; + \sum_{1<j\leq n} x_j G_{1,j}(x_1, x_j) \\ & + \sum_{1<j\leq n-1} \sum_{j<k\leq n} x_j x_k G_{i,j,k}(x_1, x_j, x_k) \\ & + ... + \; x_2 ... x_n G_{1,2,...,n}(x_1, x_2, ..., x_n) \Big\} \end{aligned} \tag{2.10}$$

Since a neighbourhood structure should be implicit in the model's joint distribution, the contribution to this energy difference of terms not falling within any of the contained cliques should be zero; hence, those particular G-functions must be zero. Any other G-functions may take arbitrary form provided the positivity condition for the ensemble distribution is maintained, i.e. $U(\mathbf{x}) > -\infty, \; \forall \mathbf{x} \in \mathbf{X}$.

This theorem provides a very general basis for the specification of MRF joint distribution functions. Many have been used throughout the literature [67] but only a subset will be reviewed here. Specifically, these are the Ising and Potts models and Gaussian MRF's, all termed *auto-models* by Besag [9] since their energy functions take the general form,

$$U(\mathbf{x}) \; = \; \sum_{1\leq i\leq n} x_i G_i(x_i) \; + \; \sum_{1\leq i\leq n-1} \sum_{i<j\leq n} \beta_{i,j} x_i x_j \tag{2.11}$$

where $\beta_{i,j}$ denotes pre-defined model parameters which may be, but are not necessarily site dependent.

2.2.1 The Ising and Potts Models

To model the underlying image (or more specifically, the interactions between the site labels of the hierarchical model) a spatial process is required which will promote the clustering of identical labels between neighbouring sites on the lattice. Potts models have provided this function throughout image segmentation literature but to gain an understanding of how they function it is necessary to first examine their binary relation, the Ising Model.

The Ising Model originates from the statistical mechanical theory of phase transitions [19]. The model has wide ranging applications, including that of modelling liquid-gas phase transitions but its primary usage has been in the modelling of magnetic materials. The driving philosophy behind the Ising Model is the concept of modelling the macroscopic characteristics of a lattice material through the specification of its microscopic or inter-molecular interactions. For example, consider a rectangular lattice of N spins in the

presence of a magnetic field H. The spins at each site (or molecule) will take values from ± 1 depending on their alignment with the external magnetic field. However, each spin will interact with its nearest neighbours, thus giving rise to an interaction energy. These two components comprise the energy of the system, defined by,

$$E = -H\mu \sum_i \sigma_i - J \sum_{ij} \sigma_i \sigma_j \tag{2.12}$$

where μ is the magnetic moment of an individual spin and J is the coupling constant of the system. The summation over ij is defined over all nearest neighbour pairs in the lattice. The macroscopic property of interest is the magnetisation of the system, defined by $M = \mu \sum_i \sigma_i$. Experimental evidence typically shows the relationship between the external magnetic field and magnetisation of a lattice to be the well known hysteresis curve of Figure 2.1(a). However, this curve is in fact attributable to non-ideal conditions (thermodynamic equilibrium not being reached within the material) and the ideal property is shown in Figure 2.1(b) (this may be observed by using a soft iron bar which is maintained in thermodynamic equilibrium by applying a continuous mechanical disturbance). The non-differentiable segment of the curve is known as the spontaneous magnetisation of the system and exists due to a cooperative phenomenon related to the inter-molecular interactions. The spontaneous magnetic property of a physical material is caused by long range interactions between individual spins and magnetic moments which are not immediate neighbours. The applicability of the Ising model stems from its ability to model these long range correlations.

When transferring such a model to the domain of image processing two fundamental characteristics are of concern; firstly, the Ising model must be reformulated to incorporate a more general state space that simply taking values from $\{\pm 1\}$ at each lattice site since an image will typically be comprised of several different regions; secondly, the relationship between spontaneous magnetisation and temperature is critical.

The first of the issues is addressed by the use of the Potts model. This is a generalisation of the Ising Model that allows the assignment of one of a multiple number of states to each site. Potential functions[2] are defined for individual cliques and these replace the interaction term in the Ising Model.

The relationship between spontaneous magnetisation and temperature deserves more consideration. Image segmentation algorithms will typically comprise a multidimensional

[2]A potential function is defined for every clique of the model and gives the associated contribution to the model's energy function for that particular clique. See [67] or [73] for a more thorough definition.

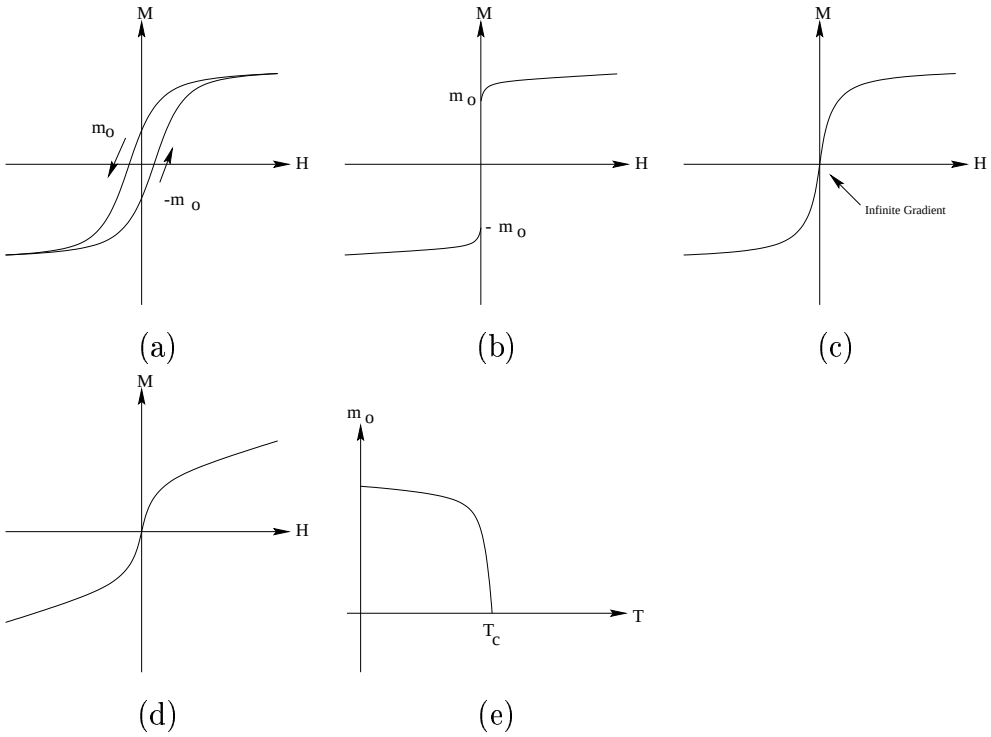
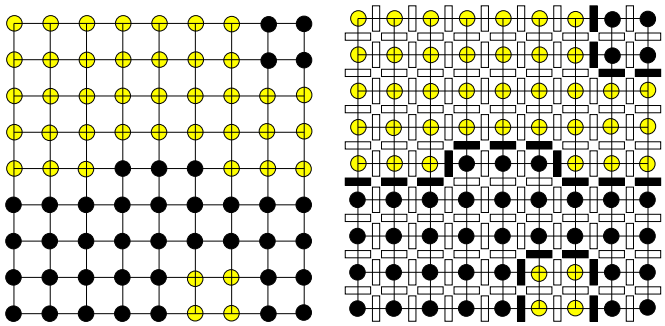


Figure 2.1: *Characteristics of the Ising Model: (a) the hysteresis effect, (b) M against H when $T < T_c$, (c) $T = T_c$, (d) $T > T_c$, (e) m_0 against T.*

optimisation process over an energy surface which is highly multi-modal. Most successful algorithms utilise some form of annealing process (see section 2.3 for a review of simulated annealing) on the associated Gibbs distribution. Inherent in such an algorithm is the steady reduction of temperature until the system settles at what is hoped to be the global energy minimum. Under consideration therefore is the increase in the Ising model's spontaneous magnetisation, or in the image processing sense, the *residual memory* of the optimisation process as temperature decreases.

Figures 2.1(b-d) demonstrate the change in the H-M characteristic as temperature is increased. The spontaneous magnetisation or non-differentiable component of the curve decreases in length until at the *critical* or Curie temperature it exists as a specific singularity on the H-M curve. Above this temperature there is no spontaneous magnetisation and so one might observe that an annealing algorithm would simply mix, allowing no long range correlations to propagate throughout the image model, hence implying no memory in the optimisation process.

The relationship between temperature and spontaneous magnetisation is often similar



(a) (b)

Figure 2.2: *A common realisation of a binary line process: (a) the current configuration of states, (b) a suitable realisation of a line process given the configuration of states.*

to that shown in Figure 2.1(e). It may be observed that at the critical temperature T_c the graph departs from the axis. Thus, when considering an annealing algorithm at this temperature and below, it is the mixing deficiences inherent in the sampling process that will limit the probability of convergence to the global minimum. It is for this reason that methods from both renormalisation group theory and auxiliary variable Markov Chain Monte Carlo (MCMC) (these will be elaborated upon later) have been employed so successfully in image processing problems; both of these methodologies improve mixing by increasing the speed by which long range interactions can affect a large change from the current realisation of the model.

2.2.2 Line Processes

The prior on the underlying segmentation (previously denoted $p(\mathbf{x} \mid \boldsymbol{\theta}^{(X)})$ in section 2.1) effectively provides a smoothing process. The Potts model, described in the previous section, has often been used to perform this role. However, for particular problems, where images have strong discontinuities, the Potts model may cause *oversmoothing* [38]. To overcome this problem Geman & Geman [39] proposed combining the underlying label MRF with an additional, 'line process'.

A line process comprises a lattice of random variables $\mathbf{l} \in \mathbf{L}$, whose sites corresponded with vertical and horizontal boundaries between adjacent pixels of the image lattice (hence an $M \times N$ image lattice will require a $2M-1 \times 2N-1$ line process; see figure 2.2). Hence the supervised segmentation problem, originally given by equation 2.1 can be reformulated

as,

$$\hat{\mathbf{x}}_{MAP}, \hat{\mathbf{l}}_{MAP} = \arg \max_{\mathbf{x}\in\mathbf{X},\ \mathbf{l}\in\mathbf{L}} p(\mathbf{x} \mid \boldsymbol{\theta}^{(X)}, \mathbf{l})\ p(\mathbf{l} \mid \boldsymbol{\theta}^{(L)}) \prod_{c\in\boldsymbol{\Lambda}} p\big(\{\mathbf{y}_s, s : x_s = c\} \mid \boldsymbol{\theta}_c^{(Y)}\big) \quad (2.13)$$

This formulation of the optimisation problem makes the Gibbs prior on the segmentation conditional on the configuration of the line process. The inclusion of a line process in the image segmentation problem, as defined in the above equation, will also allow a spatially dependent prior to be imposed on that line process. The resulting MAP estimation problem can therefore be defined using a Gibbs posterior distribution whose prior energy function is given by

$$U(\mathbf{x}, \mathbf{l} \mid \boldsymbol{\theta}) = U(\mathbf{x} \mid \boldsymbol{\theta}^{(X)}, \mathbf{l}) + U(\mathbf{l} \mid \boldsymbol{\theta}^{(L)}) \quad (2.14)$$

$$= \sum_s \sum_{r\in\boldsymbol{\eta}_s} (1 - l_{s,r}) V_X(x_s, x_r) + \sum_{t\in\boldsymbol{\rho}_s} V_L(l_s, l_t) \quad (2.15)$$

where the line process takes values from $\{0, 1\}$ and $V_X(\cdot,\cdot)$ and $V_L(\cdot,\cdot)$ are the potential functions for the segmentation and line processes, defined for neighbourhood structures $\boldsymbol{\eta}$ and $\boldsymbol{\rho}$, respectively. The prior on the line process is often selected to emphasise continuous lines, to reject spurious edge elements, and only to accept corners and line ends with low probability. Such a model has the desirable property that it promotes structure within the image without causing oversmoothing, however, for obvious reasons horizontal and vertical boundaries will be inherently favoured. It is possible to overcome this problem by generalising the binary line process into a multilevel process where each state is indicative of the corresponding orientation of the boundary element. There is however, one further and more serious drawback to the usage of line processes; they introduce many arbitrary hyper-parameters $\boldsymbol{\theta}^{(L)}$ into the model, all of whose MAP estimates are difficult to obtain. Consequently the use of line processes is generally restricted throughout the literature, to processes whose hyper-parameters can be set *a priori*.

The fundamental concepts underlying a line process were further addressed by Geman & Reynolds [38]. When considering the problem of image restoration (and possibly that of segmentation) it becomes apparent that the potential function $V_X(\cdot,\ \cdot)$ of equation 2.15, can be expressed as a single cost measure $V_X(x_s, x_r) = \phi(x_s - x_r)$ on the label differences. For segmentation this introduces a slight difficulty since the relationship between label value and observed gray-scale is unlikely to be linear. In fact in the segmentation case the cost might typically be given by $\phi(x_s - x_r) = 1 - 2\delta(x_s, x_r)$, where $\delta(\cdot)$ is the Kronecker delta function.

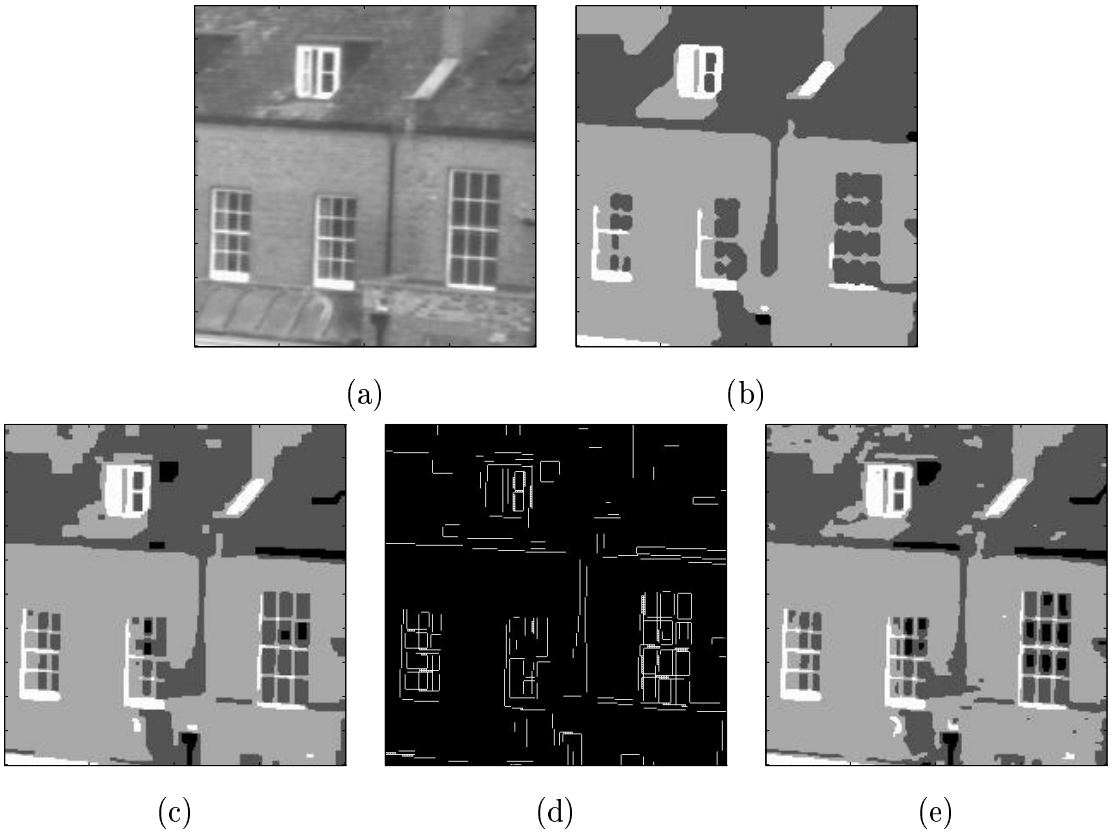


Figure 2.3: *Segmentations using various line process models (a) original image, (b) segmentation without line process, (c) segmented image with line process, (d) the corresponding line process, (e) segmentation using the modified potential function of Geman & Reynolds.*

However, if the cost function is replaced by a similar, but continuously differentiable variant, for example $\phi^*(x_s - x_r) = \left(\frac{x_s - x_r}{\Delta}\right)^2 - 1$, then in the absence of any prior term on the line process, optimisation over both $\mathbf{L}$ and $\mathbf{X}$ becomes equivalent to simply optimising over $\mathbf{X}$, but using a different energy function: specifically,

$$\inf_L \sum_s \sum_{r \in \eta_s} (1 - l_{s,r}) \phi^*(x_s - x_r) = \sum_s \sum_{r \in \eta_s} \left(\frac{x_s - x_r}{\Delta}\right)^2 - 1 \tag{2.16}$$

where Δ is a pre-specified constant.

Geman & Reynolds [38] extended this result for more general ϕ^*. To see this first define the notation $b_{s,r} = 1 - l_{s,r}$. Next define $u_{s,r} = \frac{x_s - x_r}{\Delta}$ and then redefine the prior energy attributable to each clique as $\psi(b_{s,r})$. After dropping the subscripts from the notation for

convenience, the new cost function $\phi(u)$ will be given by

$$\phi(u) \;=\; \inf_{0 \le b} \; bu^2 \;+\; \psi(b) \tag{2.17}$$

The above equation states that the desired function will be the infinum of a family of Gaussians. Although $\phi(u)$ is the required function, it is important to realise the significance of $\psi(b)$, since this effectively defines any prior knowledge of the line process. Desirable properties for this prior might be $\psi(0) = 0$ and that $\psi(b)$ is monotonically decreasing. Theorem 1 of [38] states that $\phi(u)$ will always exist for a $\psi(b)$ with these properties and will itself have further desirable properties, specifically; $\phi(0) = -1$, $\phi(\sqrt{u})$ is concave and $\lim_{u\to\infty} \phi(u) = 0$.

To summarise, all that is necessary to incorporate one of the fundamental properties of a line process, in particular to provide the ability to model rapid transitions in gray level, is to use a function that satisfies the above properties for $\phi(u)$, and to check that the associated $\psi(b)$ gives a desirable prior distribution for the related line process. Geman & Reynolds [38] suggest using the function $\phi(u) \;=\; -\frac{1}{1+u^2}$, giving a prior on the line process of $\psi(b) = b - 2\sqrt{b},\ \forall b : 0 \le b \le 1$.

As with the full line process described previously, the model just derived will introduce a bias towards vertical and horizontal edges. However, Hurn & Jennison [51] showed that this problem may be overcome by structuring the model to include diagonal cliques.

The effect of using Geman & Reynolds' process [38] is compared in Figure 2.3 with that when using either the standard Potts model or a Potts model with a full line process of the type described in [39]. The modelling of sharp transitions or edges in gray-scale is clearly better when using both the line process and the modified potential $\phi(u)$, but the difference between these two is not so clear. However, for ease of implementation the modified potential is far superior, while the full line process engenders possibilities of applying higher level image processing procedures using the line process as input.

2.2.3 The Gaussian Markov Random Field Model

The Gaussian Markov Random Field (GMRF) is the most widely used process for the modelling of various natural and man-made textures. Its effectiveness has been demonstrated through numerous classification and reconstruction experiments, for example: Chellappa [54] showed the range of textures that could be generated from GMRF models and derived a feature vector, which included spectral features, for classification purposes; Chellappa &

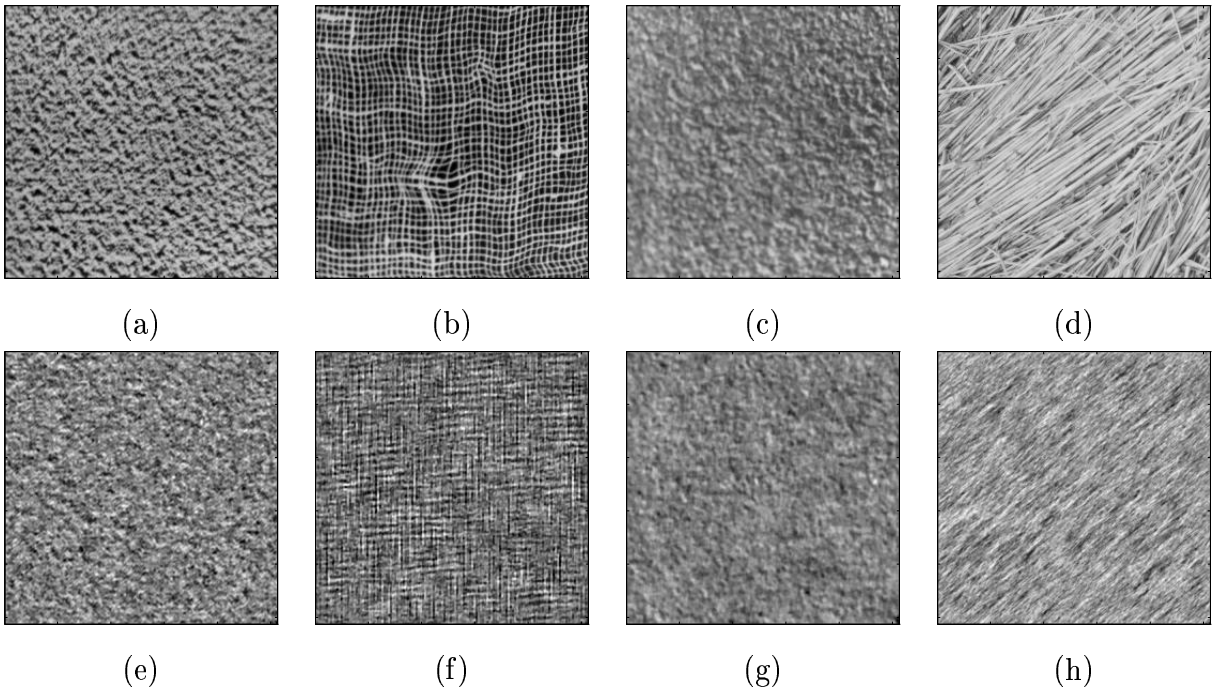


Figure 2.4: *GMRF modelling of Brodatz micro-textures: (a-d) original images corresponding to paper, burlap, cork and straw; (e-h) reconstruction using GMRF maximum-likelihood parameter estimates over a 44 pixel neighbourhood.*

Chatterjee [21] [20] used least squares estimates of GMRF parameters to classify Brodatz textures [16]; Cohen, Fan & Attali [25] used GMRF's for the automated inspection for flaws of textile fabrics and finally figure 2.4 demonstrates the effectiveness of using GMRF's to reconstruct Brodatz micro-textures using an algorithm of the type given in [2].

The GMRF is specified on an $M \times N$ image lattice $\mathbf{\Omega}$, indexed by a vector or pair $\mathbf{s} = [s_1, s_2]^T$, such that $\mathbf{\Omega} = \{\mathbf{s}; 1 \leq s_1 \leq M, 1 \leq s_2 \leq N\}$. The GMRF comprises a Gauss-Markov process which may be represented as a non-causal auto-regressive (AR) process,

$$y_s = \sum_{\tau \in \rho} \theta^{(\tau)} y_{s+\tau} + e_s \tag{2.18}$$

where $\{\theta^{(\tau)} : \tau \in \rho\}$ is the set of correlation coefficients associated with the set of translations from the central pixel which define the neighbourhood structure, i.e. $\{\tau = [\tau_1, \tau_2]^T; \tau \in \rho\}$, and e_s is a zero mean Gaussian noise process with autocorrelation given

by

$$E[e_s e_{s+\tau}] = \begin{cases} \sigma^2, & \tau = \mathbf{0} \\ -\theta^{(\tau)}\sigma^2, & \tau \in \rho \\ 0, & otherwise \end{cases} \tag{2.19}$$

Alternatively, the process may be written in terms of a linear matrix equation [56] $\mathbf{R}_{\boldsymbol{\theta}}\mathbf{y} = \mathbf{e}$. Thus, the covariance matrix of the noise process $\mathbf{e}$ is given by $\sigma^2\mathbf{R}_{\boldsymbol{\theta}}$ and it can be shown [56] that the covariance of the process $\mathbf{y}$ is given by $\boldsymbol{\Sigma} = \sigma^2\mathbf{R}_{\boldsymbol{\theta}}^{-1}$. The joint distribution for the GMRF may therefore be written

$$p(\mathbf{y} \mid \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{MN/2} \ |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}\, \mathbf{y^T}\boldsymbol{\Sigma}_{\mathbf{c}}^{-\mathbf{1}}\mathbf{y} \right\} \tag{2.20}$$

$$= \frac{|\mathbf{R}_{\boldsymbol{\theta}}|^{\frac{1}{2}}}{(2\pi\sigma^2)^{MN/2}} \exp\left\{ -\frac{1}{2\sigma^2}\, \mathbf{y^T}\mathbf{R}_{\boldsymbol{\theta}}\mathbf{y} \right\} \tag{2.21}$$

With no loss of generality it is possible to write $\theta^{(\tau)} = \theta^{(-\tau)}$ thus ensuring the symmetry of the correlation matrix $\mathbf{R}_{\boldsymbol{\theta}}$. The GMRF exhibits a Markov property: the process is Markov with respect to a neighbourhood structure, thus satisfying the Hammersley-Clifford theorem,

$$p(y_s \mid \mathbf{y}_{\rho_s}\sigma, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}\Big(y_s - \sum_{\tau:\tau\in\rho} \theta^{(\tau)}\big(y_{s+\tau} + y_{s-\tau}\big)\Big)^2 \right\} \tag{2.22}$$

The quadratic term of equation 2.21 may be simplified [86] $\mathbf{y}^T\mathbf{R}_{\boldsymbol{\theta}}\mathbf{y} = c_0 - \boldsymbol{\theta}^T\mathbf{c}$, where the statistics c_0 and the vector $\mathbf{c}$, indexed by τ, are given by,

$$c_0 = \textstyle\sum_{s\in\Omega} y_s^2, \qquad c(\tau) = \sum_{s\in\Omega} y_s y_{s+\tau} : \tau \in \rho \tag{2.23}$$

Using the Neyman-Fisher factorisation theorem [8] Chellappa and Chatterjee [21] showed these statistics formed a lossless feature set for texture classification.

For the covariance matrix of the GMRF to be positive definite, the correlation parameters must satisfy the condition $1 - \boldsymbol{\theta}^T\boldsymbol{\phi}_s > 0$, where each element of $\boldsymbol{\phi}_s$ is given by $\cos(\frac{2\pi s_1\tau_1}{M} + \frac{2\pi s_2\tau_2}{N})$. This follows directly from re-writing the determinant of the correlation matrix as a product function [56];

$$|\mathbf{R}_{\boldsymbol{\theta}}| = \prod_{s\in\Omega}(1 - \boldsymbol{\theta}^T\boldsymbol{\phi}_s) \tag{2.24}$$

The two-dimensional power spectrum $S_Y(\omega)$ has been shown by Woods [97] to be given by

$$S_Y(\omega) = \frac{\sigma^2}{1 - \sum_{\tau\in\rho}\theta_\tau \cos\left(\frac{2\pi}{M}\tau_1\omega_1 \ + \ \frac{2\pi}{N}\tau_2\omega_2\right)} \tag{2.25}$$

and the inverted elements $S_Y(\omega)^{-1}$ form the eigenvalues of the inverse covariance matrix $\mathbf{\Sigma}^{-1} = \mathbf{R}_{\boldsymbol{\theta}}\sigma^{-2}$, see [65] or [57]. The related autocorrelation function is therefore

$$R_Y(s) = \frac{1}{MN}\sum_{\omega\in\Omega} S_Y(\omega)\cos\left(\frac{2\pi}{M}s_1\omega_1 + \frac{2\pi}{N}s_2\omega_2\right) \tag{2.26}$$

$$= \frac{\sigma^2}{MN}\sum_{\omega\in\Omega}\frac{\cos\left(\frac{2\pi}{M}s_1\omega_1 + \frac{2\pi}{N}s_2\omega_2\right)}{1-\sum_{\tau\in\rho}\theta_\tau\cos\left(\frac{2\pi}{M}\tau_1\omega_1 + \frac{2\pi}{N}\tau_2\omega_2\right)} \tag{2.27}$$

A derivative of the GMRF, the Compound Gaussian Markov Random Field (CGMRF) [53] incorporates a number of line processes, dependent on the neighbourhood structure of the GMRF. These effectively turn on or off the interaction between neighbouring pixels, thus allowing discontinuities to occur within a textured image. The CGMRF density function is derived from that of the GMRF model, defined in equations 2.20 and 2.21. If for example the GMRF is defined using a nearest neighbourhood structure, then the CGMRF may be derived by incorporating a line process $\mathbf{l}$ that specifies interactions between both vertically and horizontally adjacent sites. The conditional distribution for a pixel at site s now becomes

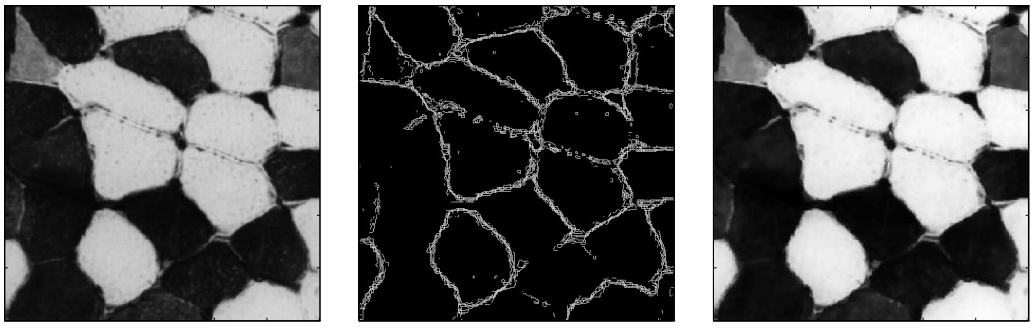
$$p(y_s \mid \mathbf{y}_{\rho_s}\mathbf{l},\sigma,\boldsymbol{\theta}) \propto \exp\Big\{-\frac{1}{2\sigma^2}\sum_{\tau\in\rho}\theta^{(\tau)}[(y_s-y_{s+\tau})^2(1-l_{s,s+\tau})+(y_s-y_{s-\tau})^2(1-l_{s-\tau,s})] - \frac{1}{2\sigma^2}\left(1-2\sum_{\tau\in\rho}\theta^{(\tau)}\right)y_s^2\Big\} \tag{2.28}$$

An exponential distribution is typically assumed for the label process prior, i.e. $p(l_s^{(\tau)}) \propto \exp\{-\beta l_s^{(\tau)}\}$, resulting in a joint distribution across the image and line process lattices given by,

$$p(\mathbf{y},\mathbf{l}\mid\sigma,\boldsymbol{\theta},\beta) = \frac{1}{Z(\sigma,\boldsymbol{\theta},\beta)}\exp\Big\{\sum_{s\in\Omega}\Big\{-\frac{1}{2\sigma^2}\sum_{\tau\in\rho}\theta^{(\tau)}(y_s-y_{s+\tau})^2(1-l_{s,s+\tau}) - \frac{1}{2\sigma^2}\left(1-2\sum_{\tau\in\rho}\theta^{(\tau)}\right)y_s^2 + \beta l_s^{(\tau)}\Big\}\Big\} \tag{2.29}$$

The fitting of such models to data is discussed more extensively in section 2.5.4, but the basic methodology requires the incorporation of the above equation into the model distribution as a prior, then forming the posterior distribution by using an isotropic Gaussian likelihood function to link the observed data $\mathbf{z}$ to the image model $\mathbf{y}$:

$$p(\mathbf{y},\mathbf{l}\mid\sigma,\boldsymbol{\theta},\beta,\mathbf{z}) \propto p(\mathbf{z},\mid\mathbf{y},\rho)\; p(\mathbf{y},\mathbf{l}\mid\sigma,\boldsymbol{\theta},\beta) \tag{2.30}$$



(a) (b) (c)

Figure 2.5: *CGMRF modelling of muscle fibres: (a) original image corresponding* $\mathbf{z}$ *of equation 2.30, (b) fitting of the line process,* $\mathbf{l}$*, (c) fitting of the process,* $\mathbf{y}$*.*

where $p(\mathbf{z},\mid \mathbf{y},\lambda^2) \propto \exp\left\{ -\frac{1}{2\lambda^2}\sum_{s\in\Omega}(z_s - y_s)^2 \right\}$. An example of the use of such a model is given in figure 2.5. Figure 2.5(a) shows the original cross-sectional image of some muscle fibres while (b-c) show the resulting fitting of both the underlying process and line processes. An algorithm of the type described later in section 2.5.4 was used, combining estimation of these processes with that of the model parameters.

2.2.4 GMRF Relationship to Simultaneous Autoregressive Models

The simultaneous autoregressive (SAR) process is described by the following equation,

$$y_s \;=\; \sum_{\tau\in\rho} \theta^{(\tau)} y_{s+\tau} + \sqrt{\nu}\; w_s \tag{2.31}$$

where ν is a variance parameter relating to the independent and identically distributed zero mean and unit variance noise process w_s. The process is bilateral Markov unless the process is strictly auto-regressive, then it is Markov. The major difference between the GMRF and SAR models is the correlation structure or otherwise of the noise process; $\mathbf{e}$ of equation 2.18 is correlated, while $\mathbf{w}$ in equation 2.31 is not.

The SAR model can be related to the GMRF via the Karhunen-Loeve transform. Recalling that the GMRF may be expressed as $\mathbf{R}_{\boldsymbol{\theta}}\mathbf{y} = \mathbf{e}$, it is possible to decorrelate the noise vector $\mathbf{e}$ by pre-multiplication of the image vector by the square root of the correlation matrix, i.e $\sqrt{\mathbf{R}_{\boldsymbol{\theta}}}$. Eigen-decomposition of $\mathbf{R}_{\boldsymbol{\theta}}$ allows an equivalent realisation of the GMRF to be written in terms of an uncorrelated noise vector $\mathbf{w}$, with variance ν:

$$\sqrt{\mathbf{R}_{\boldsymbol{\theta}}}\;\mathbf{y} \;=\; \sqrt{\nu}\;\mathbf{w} \tag{2.32}$$

where

$$\sqrt{\mathbf{R}_{\boldsymbol{\theta}}} \;=\; \sqrt{\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T} \;=\; \sum_{i=1,..,MN} \mathbf{u}_i \mathbf{u}_i^T \sqrt{\lambda_i} \tag{2.33}$$

Here $\mathbf{U}$ is the matrix of eigenvectors of $\mathbf{R}_{\boldsymbol{\theta}}$, specifically $\{\mathbf{u}_i,\ i = 1, ..., MN\}$, and $\boldsymbol{\Lambda}$ is the matrix whose diagonal elements are the associated eigenvalues $\{\lambda_i, i = 1, ..., MN\}$. The noise process $\mathbf{e}$ may therefore be expressed in terms of uncorrelated noise: $\mathbf{e} = \sqrt{\mathbf{R}_{\boldsymbol{\theta}}}\ \mathbf{w}$. Equation 2.32 does not necessarily describe a SAR process, however under specific conditions (see [57]) relating to the factorability of the two-dimensional z-transform of the difference equation 2.18, the GMRF may indeed be re-written as an SAR process. The converse always holds: any SAR models may be re-expressed in GMRF form [56].

2.3 Segmentation Techniques

The problem of segmenting an image using a Markov Random Field (MRF) model is difficult. This stems from the need to find the segmentation that yields either the Maximum *a posteriori* or the maximiser of posterior marginals (MPM) estimate of a multidimensional Gibbs distribution (as outlined in the introduction to this chapter, section 2.1).

It comes as no surprise therefore, that probably the most computationally efficient and hence most widely used segmentation algorithm makes a massive, *ad hoc* simplification to the problem: this is the Iterative Conditional Modes (ICM) algorithm of Besag [11]. Besag argues, "*suppose* $\hat{\mathbf{x}}$ *denotes a provisional estimate of the true scene* $\mathbf{x}^*$ *and that our aim is merely to update the current colour* $\hat{x}$ *at pixel* i *in the light of all available information. Then a plausible choice is the colour which has maximum conditional probability, given records* y *and the current construction* $\hat{\mathbf{x}}_{\boldsymbol{\Omega}\setminus i}$ *elsewhere.*"

Applying Bayes' theorem to the segmentation problem and adopting the notation defined in the introduction to this chapter, i.e. section 2.1, the ICM algorithm relies on updating sites in the lattice in a specific order, according to,

$$\hat{x}_s \;=\; \arg\max_{x_s \in \Lambda}\ p(x_s \mid \mathbf{y}, \hat{\mathbf{x}}_{\boldsymbol{\Omega}\setminus i}, \boldsymbol{\theta}) \tag{2.34}$$

$$\;=\; \arg\max_{x_s \in \Lambda}\ p(y_s \mid x_s, \boldsymbol{\theta}^{(Y)}) p(x_s \mid \hat{\mathbf{x}}_{\boldsymbol{\Omega}\setminus i}, \boldsymbol{\theta}^{(X)}) \tag{2.35}$$

Since all updates must inherently either increase or maintain the joint probability of

the current realisation, the algorithm must by definition converge to a local maximum. However, there is no guarantee of convergence to the global mode.

A similar algorithm was applied to the problem of unsupervised texture segmentation by Cohen & Cooper [23]. Using a hierarchical model with individual textures modelled by GMRF and using a Potts' model as a prior on the label field, a process similar to the ICM algorithm was used as an optimiser. However, Besag's [9] coding method for parameter estimation was modified to provide a mechanism to improve speed of convergence: if an $M \times N$ image lattice $\mathbf{\Omega}$ is divided in two sets of sites whose corresponding random variables are mutually independent, i.e. $\mathbf{\Omega}_1 = \{(i,j); \text{if } j \text{ is odd}, i = 1,3,5,...,M-1, \text{else } i = 2,4,6,...,M,\ 1 \leq j \leq N\}$ and $\mathbf{\Omega}_2 = \{(i,j); \text{if } j \text{ is odd}, i = 2,4,6,...,M, \text{else } i = 1,3,5,...,M-1,\ 1 \leq j \leq N\}$, then the sites corresponding to subset $\mathbf{\Omega}_1$ may be updated in parallel from probability distributions conditioned on the sites of $\mathbf{\Omega}_2$; next the sites of $\mathbf{\Omega}_2$ can be updated conditioned on $\mathbf{\Omega}_1$.

Geman & Geman [39] introduced a more sophisticated optimisation algorithm to image segmentation, known as simulated annealing. The methodology solved the MAP estimation problem, specified earlier in equation 2.1. Simulated Annealing draws heavily upon ideas from statistical mechanics and in particular, those surrounding the Gibbs distribution. The optimisation criterion may be any of the following,

$$\begin{aligned} \hat{\mathbf{x}} &= \arg\max_{\mathbf{x}\in\mathbf{X}} \ p(\mathbf{x}) \\ &= \arg\max_{\mathbf{x}\in\mathbf{X}} \ \frac{1}{Z}\exp\Big\{-\frac{1}{T}U(\mathbf{x})\Big\} \\ &= \arg\min_{\mathbf{x}\in\mathbf{X}} \ U(\mathbf{x}) \end{aligned} \tag{2.36}$$

The final equation, minimising over the energy surface suggests the essence of the problem: the energy surface is likely to be non-convex, in fact highly multi-modal, hence any deterministic optimisation algorithm is likely to settle at a local minimum.

Simulated annealing has been considered to be a methodology with the potential to overcome this problem. First a cooling schedule is designated for the Gibbs distribution temperature parameter. The temperature begins in a hot state, causing the energy surface to be relatively flat, since the system is relatively excited. The temperature is then steadily lowered, making the local and global minima become further pronounced. Then as the temperature tends to zero, the surface moves towards a set of inverted peaks, the deepest at the global minimum.

Interestlingly, there occurs a direct link between the Iterative Conditional Modes algo-

rithm of Besag [11] and Geman & Geman's [39] simulated annealing algorithm: ICM is simply simulated annealing but with the temperature set initially and then held at zero, in other words, the system is instantaneously frozen.

To implement the simulated annealing algorithm still requires a method of exploring the energy surface. Geman & Geman [39] provided the Gibbs Sampler to achieve this. The algorithm uses a Markov Chain Monte Carlo technique to facilitate the sampling of the multidimensional Gibbs distribution $\pi(\mathbf{x}) = \frac{1}{Z}\exp\{\frac{1}{T}U(\mathbf{x})\}$, defined on the rectangular lattice $\mathbf{\Omega}$. The Gibbs sampler works by updating each random variable individually, but conditional on the states of the surrounding sites. Geman & Geman's [39] Theorem A states that, *if for each* $s \in \Omega$ *the sequence, indexed by* t (of updated sites) *contains* s *infinitely often, then for every starting configuration* $\boldsymbol{\chi} \in \mathbf{X}$ *and every* $\mathbf{x} \in \mathbf{X}$,

$$\lim_{t\to\infty} p(\mathbf{x}(t) = \mathbf{x} \mid \mathbf{x}(0) = \boldsymbol{\chi}) = \pi(\mathbf{x}) \tag{2.37}$$

Hence, given a regular scanning sequence, for example raster scanning, the samples drawn by the Gibbs sampler will tend to the target distribution as $t \to \infty$.

Their second theorem (Theorem B of [39]) proves convergence for simulated annealing towards the global minimum of the Gibbs distribution. If Δ is defined to be the difference between the global maximum and minimum of the energy surface $U(\mathbf{x})$, and N is the number of sites, then convergence is proved given two conditions; infinite time or iterations for the Gibbs sampling algorithm and a decreasing temperature schedule which meets the condition $T(t) \geq \frac{N\Delta}{\log t}$, for $t \geq t_0 \geq 2$. Unfortunately, such a schedule is of little practical use since it is firstly, impossible to calculate Δ and secondly, would require too many iterations to descend over a sufficient range of temperature.

Many authors have carried out research in an attempt to find an in some sense optimal annealing schedule. Much of this work has been adequately reviewed in both [77] and [75]. For example, in addition to Geman & Geman's [39] logarithmic schedule just described, there are several other fixed schedules: geometric schedules, given by $T(t) = \alpha^t T_0$, or $T(t+1) = \alpha T(t)$, for $0 \leq \alpha \leq 1$; exponential schedules, where $T(t) = T_0 \exp\{-\alpha t^\delta\}$, and where α and δ are problem specific positive constants; and finally, linear schedules, here $T(t) = (1+\alpha_1)^{\alpha_2(1-\frac{t}{N})}$ and α_1, α_2 are positive constants, while N is the total number of iterations.

Adaptive annealing schedules have also been extensively explored (again, see both [77] and [75] for details). The general motivation behind such schemes is to force the annealing algorithm to concentrate the greatest proportion of its iterations at the system's critical

temperature.

Szu & Hartley [91] suggest an alternative approach to improving the performance of the annealing algorithm. Although using Geman & Geman's [39] logarithmic cooling schedule they propose using heavy tailed proposal densities, for example the Cauchy distribution, as their Metropolis-Hastings proposal distribution. Such a procedure attacks the problem from a new angle, attempting to improve the mixing of the proposal distribution sampling algorithm, rather than developing problem specific annealing schedules. A similar philosophy underpins the more complex algorithms developed throughout chapters 4 and 5 of this thesis.

2.4 Multi-Resolution Segmentation

It is well known that Markov Random Field models (MRF's) express global relationships in terms of local statistics. Thus most segmentation or sampling schemes operate at the local level, updating sites iteratively, inducing long range interactions at the expense of much computational burden. Multi-resolution techniques attempt to circumvent this problem by providing a mechanism by which longer range interactions can quickly propagate through the image model. The importance of long range interactions to Markov Random Field theory, and particularly to image segmentation algorithms is a seminal concept of this dissertation, thus multi-resolution approaches are of manifest importance to this review.

The multi-resolution approach generally adopted begins by establishing the model parameters *a priori* at the highest resolution which is typically that of the original image. The image is then repeatedly down sampled or '*coarsened*' using some form of transform, while fresh model parameters are estimated at each of these ensuing resolutions. Segmentation is first carried out at the coarsest resolution through the use of a standard segmentation algorithm. The resulting segmentation data is then used to initialise or constrain the segmentation process at the next coarsest level. The scheme is repeated at each resolution until a segmentation is achieved at the highest and original resolution.

2.4.1 Renormalisation Group Theory

Multi-resolution algorithms have been derived from the statistical mechanical principles of renormalisation group theory. The early objective of renormalisation group theory [19] was to make possible the estimation of partition functions, thus facilitating the study of phase transitions in various materials. The concepts behind renormalisation group theory can be inferred by considering a one-dimensional Ising model in the absence of an external field. If $\sigma_1, \sigma_2, ...$ represent the spins of a system with coupling constant J, then the partition function may be expressed

$$Z_N(J,T) \;=\; \sum_{\sigma_1,\sigma_2,\ldots,\sigma_N} \exp\{\beta(\sigma_1\sigma_2 + \sigma_2\sigma_3 + \sigma_3\sigma_4 + ...)\} \tag{2.38}$$

where $\beta = \frac{J}{kT}$, k is the Boltzmann constant and T the temperature. By factoring this expression and then summing over all the evenly indexed spins, the number of degrees of freedom assigned to the model will be fractionalised:

$$\begin{aligned} Z_N(J,T) \;=\; \sum_{\sigma_1,\sigma_3,\ldots,\sigma_{N-1}} & \left[\exp\{\beta(\sigma_1+\sigma_3)\} + \exp\{-\beta(\sigma_1+\sigma_3)\}\right] \\ & \times \left[\exp\{\beta(\sigma_3+\sigma_5)\} + \exp\{-\beta(\sigma_3+\sigma_5)\}\right] \times .... \end{aligned} \tag{2.39}$$

The next goal is to re-write this expression in an identical form to that of the original Ising model, but now with only $\frac{N}{2}$ spins. To achieve this define both a function $f(J,T)$, and a new coupling constant, then form the translation operation

$$\exp\{\beta(\sigma_i+\sigma_j)\} + \exp\{-\beta(\sigma_i+\sigma_j)\} \;=\; f(J,T)\;\exp\{\beta'\sigma_i\sigma_j\} \tag{2.40}$$

where $\beta' = \frac{J'}{kT}$. To solve for these unknowns is simple (by considering the all possible values for σ_i and σ_j) and the resulting expressions are $f(J,T) = 2\cosh^{\frac{1}{2}}(2\beta)$ and $\beta' = \frac{1}{2}\log\cosh(2\beta)$. Using these functions a recursive relationship between the partition functions may be established:

$$\begin{aligned} Z_N(J,T) \;&=\; \sum_{\sigma_1,\sigma_3,\ldots,\sigma_{N-1}} f(J,T)\;\exp\{\beta'\sigma_1\sigma_3\}\;f(J,T)\;\exp\{\beta'\sigma_3\sigma_5\}\;.... \\ &=\; f(J,T)^{\frac{N}{2}}\;Z_{\frac{N}{2}}(J',T) \end{aligned} \tag{2.41}$$

Such a function has been shown to exist for other models and in particular, for two-dimensional lattice based Ising models (see [19]), but to observe how such an approach might be useful in image segmentation the procedure must be generalised [41] [40].

Consider an image segmentation or restoration problem given an observed image $\mathbf{y}$. This requires the estimation of an underlying image $\mathbf{x}$ by maximisation of its posterior

distribution $p(\mathbf{x} \mid \mathbf{y}) = \frac{1}{Z}\exp\{-H(\mathbf{x} \mid \mathbf{y})\}$. If the image is defined on a lattice $L^{(1)}$ then associate with this a Hamiltonian $H^{(1)} = H(\mathbf{x} \mid \mathbf{y})$. The renormalisation of this model will therefore involve an iterative coarsening, $L^{(1)} \rightarrow L^{(2)} \rightarrow ... \rightarrow L^{(N)}$, and a renormalisation group transformation which gives a consequent reassignment of the Hamiltonians $H^{(1)} \rightarrow H^{(2)} \rightarrow ... \rightarrow H^{(N)}$. Additionally, for convergence criteria to be satisfied, constants related to Gibbs distribution temperature are introduced into each Hamiltonian, specifically $T_1, T_1, ..., T_N$, thus the progression from fine to coarse Hamiltonian is written $H^{(1)}_{T_1} \rightarrow H^{(2)}_{T_1,T_2} \rightarrow ... \rightarrow H^{(n)}_{T_1,...T_n}$, where $H^{(1)}_{T_1} = \frac{1}{T_1}H(\mathbf{x} \mid \mathbf{y})$.

To derive recursive expressions for these Hamiltonian necessitates the identification of a conditional probability relating realisations of the process $\mathbf{x}$ on different lattices. Thus, by defining the conditional probability $p^{(n)}(\mathbf{x}^{(n)} \mid \mathbf{x}^{(n-1)})$ and integrating out the coarse lattice values, the Hamiltonians can be related by the equation,

$$\exp\Big\{ - H^{(n)}_{T_1,...T_n}(\mathbf{x}^{(n)})\Big\} = \sum_{\mathbf{x}^{(n-1)}} p^{(n)}(\mathbf{x}^{(n)} \mid \mathbf{x}^{(n-1)}) \exp\Big\{ - \frac{1}{T_n} H^{(n-1)}_{T_1,...T_{n-1}}(\mathbf{x}^{(n-1)})\Big\} \quad (2.42)$$

Typically, $p^{(n)}(\mathbf{x}^{(n)} \mid \mathbf{x}^{(n-1)})$ will be selected to be a product of Kronecker delta functions across the lattices, i.e. $p^{(n)}(\mathbf{x}^{(n)} \mid \mathbf{x}^{(n-1)}) = \prod_{i \in L^{(n)}} \delta_{x_i^{(n)} x_i^{(n-1)}}$.

Given these conditions a generalised renormalisation group algorithm may now be defined consisting of the following steps;

---

1. choose a coarsening process, define a set of conditional probabilities $p^{(n)}(\mathbf{x}^{(n)} \mid \mathbf{x}^{(n-1)})$, select values for the constants $T_1, T_1, ..., T_N$ and calculate the set of Hamiltonians, $H^{(n)}_{T_1,...T_n}(\mathbf{x}^{(n)})$.

2. Find the global minimum $\bar{\mathbf{x}}^{(N)}$ of $H^{(N)}_{T_1,...T_N}(\mathbf{x}^{(N)})$ and set n=N.

3. Find the global minimum of $H^{(n-1)}_{T_1,...T_n-1}(\mathbf{x}^{(n-1)})$ subject to the constraint,

$$p^{(n)}(\bar{\mathbf{x}}^{(n)} \mid \mathbf{x}^{(n-1)}) = \max_{\{\mathbf{x}^{(n)}\}} p^{(n)}(\mathbf{x}^{(n)} \mid \mathbf{x}^{(n-1)}) \quad (2.43)$$

   e.g. when using delta functions, ensuring corresponding pixels between lattices remain identical.

4. If $n = 1$ stop, else $n = n - 1$ and goto 3.

The specification of temperature constants $T_1, T_1, ..., T_N$ governs the convergence of the algorithm. If the conditional probabilities $p^{(n)}(\bar{\mathbf{x}}^{(n)} \mid \mathbf{x}^{(n-1)})$ are assumed to be uni-modal

(as they self-evidently are in the Kronecker-Delta case) and the original Hamiltonian $H(x)$ is also uni-modal, then Thereom 2.1 of [41] shows that if $T_1, T_1, ..., T_N$ are chosen to be *sufficiently small* then all the respective Hamiltonians have unique global minima and the renormalisation group algorithm will find the true global minimum of $H(\mathbf{x})$. In practice the selection of temperature parameters is somewhat empirical, thus making any analysis of the algorithm's convergence properties somewhat irrelevant.

A further problem with such an algorithm concerns the difficulty in calculating the necessary Hamiltonian functions using equation 2.42. However, Gidas [41] did apply the algorithm successfully to the restoration and segmentation of binary images corrupted by binary noise, thus giving an indication of the value of directly incorporating long range interactions into the optimisation process for image segmentation algorithms.

2.4.2 GMRF Multi-resolution Models

Having explored the relationship between renormalisation group theory and multi-resolution approaches to segmentation, it is interesting to observe that authors[14] [26] [62] [63] applying such approaches to practical imaging problems have in general relaxed the constraint on the optimisation process at each level of the multi-resolution pyramid (given by equation 2.43) to produce more loosely defined algorithms, for example:

---

1. choose a coarsening process to generate the N-layer multi-resolution image pyramid and calculate the associated set of Hamiltonians, using the *a priori* defined image model as a basis.
2. Find the global minimum at the coarsest resolution, i.e. $\bar{\mathbf{x}}^{(N)}$ and set n=N.
3. Supply an interpolated realisation of $\bar{\mathbf{x}}^{(n)}$ as the starting point for an optimisation at the higher resolution $n-1$ which will yield $\bar{\mathbf{x}}^{(n-1)}$
4. If $n = 1$ stop, else $n = n - 1$ and goto 3.

The reason for this divergence from the renormalisation group theory is not at first obvious. However, the issue of convergence is paramount: since the renormalisation group approach requires the *a priori* specification of a set of temperature parameters whose values

are dependent on the Hamiltonian in question, its use for semi-unsupervised segmentation problems is somewhat preclusive.

To examine the above processes in more detail, a method for coarsening the image must first be established. There are two typically adopted approaches to lowering the resolution of a random field[65]; sampling or decimation, and block to point transforms.

To illustrate the first of these two approaches, denote the highest resolution image lattice as $\mathbf{\Omega}^{(0)}$, then the sampling transform $\mathbf{\Omega}^{(k-1)} \Rightarrow \mathbf{\Omega}^{(k)}$ is defined by $x_s^{(k)} = x_{2s}^{(k-1)}$. Alternatively the transform may be expressed in terms of the linear matrix equation $\mathbf{x}^{(k)} = \mathbf{D}_0^{(k)}\mathbf{x}^{(0)}$, where $\mathbf{D}_0^{(k)}$ is sparse and is defined as in [65]. From this notation the following properties were shown in [65] to apply to the k'th resolution model: the covariance matrix is given by $\mathbf{\Sigma^{(k)}} = [\mathbf{D}_0^{(k)}]\mathbf{\Sigma^{(0)}}[\mathbf{D}_0^{(k)}]^T$; the joint distribution is

$$p(\mathbf{y^{(k)}} \mid \mathbf{\Sigma^{(k)}}) = \frac{\mathbf{1}}{(\mathbf{2}\pi)^{\frac{\mathbf{MN}}{\mathbf{2^{2k+1}}}} \; |\mathbf{\Sigma^{(k)}}|^{\frac{\mathbf{1}}{\mathbf{2}}}} \exp\left\{ -\frac{\mathbf{1}}{\mathbf{2}} [\mathbf{y^{(k)}}]^{\mathbf{T}} [\mathbf{\Sigma^{(k)}}]^{-\mathbf{1}} [\mathbf{y^{(k)}}] \right\} \tag{2.44}$$

and the power spectrum is given by

$$S_Y^{(k)}(\omega) = \frac{1}{2^{2k}} \sum_{\mathbf{r} \in C^{(k)}} S_Y^{(0)}(\omega_1 + \frac{M}{2^k} r_1, \omega_2 + \frac{N}{2^k} r_2) \tag{2.45}$$

where $C^{(k)} = \{\mathbf{r} : 0 \leq r_1, r_2 \leq 2^k - 1\}$ and $S_Y^{(0)}(\boldsymbol{\omega})$ was given in equation 2.25. Although equation 2.44 is that of a multivariate Gaussian distribution, the random field which it represents, $\mathbf{y}^{(k)}$, is in general non-Markov. This may be elucidated by observing that the new power spectrum cannot be re-written in the form of that found for a GMRF, i.e. that of equation 2.25.

The block to point approach may be defined (see [65]) by the equation

$$x_s^{(k)} = b \sum_{r \in \mathbf{B}} x_{2s+r}^{(k-1)} + \nu v_s^{(k)} \tag{2.46}$$

where $\mathbf{B} = \{r = [r1, r2]^T : 0 \leq r1 \leq 2^k - 1, 0 \leq r2 \leq 2^k - 1\}$ and $v_s^{(k)}$ is an i.i.d. zero-mean, unit variance Gaussian noise process. Thus $x_s^{(k)}$ is Gaussian distributed with variance ν about the mean of the associated block of pixels at the higher resolution $k-1$. The corresponding joint distribution and power spectra to equations 2.44 and 2.45 are more complex than from the former case (again, see [65]).

To allow conventional GMRF segmentation or optimisation techniques to be utilised at coarser resolutions Krishnamachari & Chellappa [62] [63] suggested a method by which the non-Markov processes associated with coarser data obtained via the sampling mechanism

might be approximated by further GMRF models. To achieve this, Lakshmanan & Derin's proposal [65] to minimise the Kullback-Leibler distance [8] between the exact non-Markov model and a specified family of GMRF models was followed. Thus GMRF parameter approximations may be obtained from the expression,

$$\left([\boldsymbol{\theta}^{(k)}]^*, [\sigma^{(k)^2}]^*\right) = \arg \min_{(\boldsymbol{\theta}^{(k)}, \sigma^{(k)^2})} \mathrm{E}_p\left[\log p(\mathbf{y}) - \log p^*(\mathbf{y})\right] \tag{2.47}$$

$$= \arg \max_{(\boldsymbol{\theta}^{(k)}, \sigma^{(k)^2})} \mathrm{E}_p\left[p^*(\mathbf{y})\right] \tag{2.48}$$

$$= \arg \max_{(\boldsymbol{\theta}^{(k)}, \sigma^{(k)^2})} \left\{ \frac{1}{2} \sum_{s \in \Omega^{(k)}} \log(1 - \boldsymbol{\theta}^{(k)^T} \boldsymbol{\phi}_s^{(k)}) - \frac{MN}{2} \log \sigma^{(k)^2} - \frac{1}{2\sigma^{(k)^2}} \left[\mathrm{E}_p[c_0] - \boldsymbol{\theta}^{(k)^T} \mathrm{E}_p[\mathbf{c}]\right] \right\} \tag{2.49}$$

where c_0 and $\mathbf{c}$ were defined in equation 2.23 and $\boldsymbol{\phi}_s^{(k)}$ was defined prior to equation 2.24. Thus the approximation of the GMRF parameters merely requires the computation of moment statistics from the true non-Markov distribution and their substitution in the optimisation problem given by the previous equation.

Alternatively, the Kullback-Leibler distance between local conditional distributions can be minimised using a similar process (see [63]), resulting in analytical expressions for GMRF parameters;

$$[\boldsymbol{\theta}^{(k)}]^* = [\mathrm{E}[\ \mathbf{x}_{\rho_s} \mathbf{x}_{\rho_s}^T\ ]]^{-1}\ \mathrm{E}[\ x_s \mathbf{x}_{\rho_s}\ ] \tag{2.50}$$

$$[\sigma^{(k)^2}]^* = \mathrm{E}[x_s^2] - [\boldsymbol{\theta}^{(k)}]^* \mathrm{E}[\ x_s \mathbf{x}_{\rho_s}^T\ ] \tag{2.51}$$

where the autocorrelation terms at the k'th level may always be written in terms of those on $\boldsymbol{\Omega}^{(0)}$, $\mathrm{E}_{p^{(k)}}[\ x_s^{(k)} x_{s+r}^{(k)}\ ] = \mathrm{E}_{p^{(0)}}[\ x_{2^k s}^{(0)} x_{2^k(s+r)}^{(0)}\ ]$ and these were given in equation 2.27.

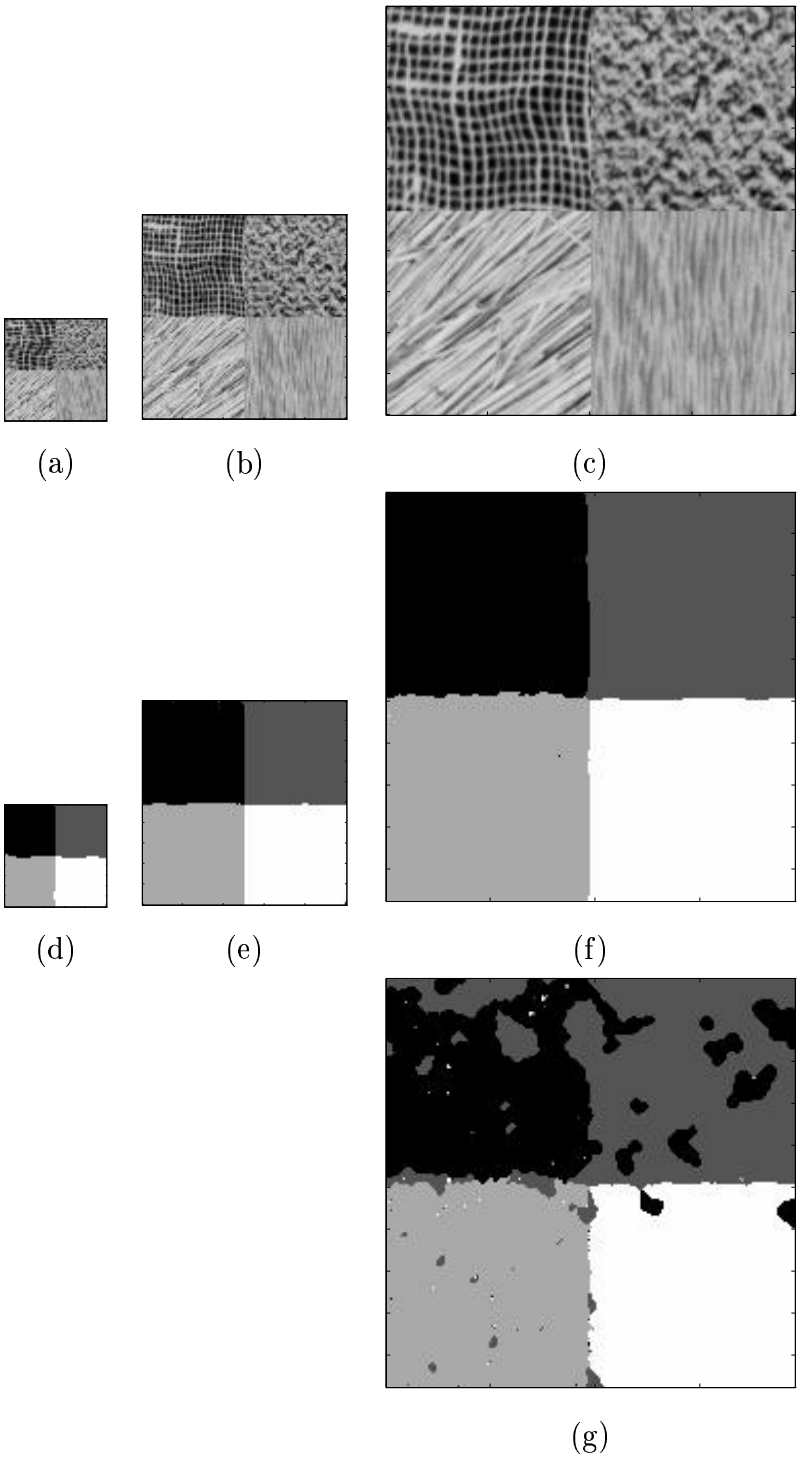


Figure 2.6: *GMRF Multiresolution Example (a-c) coarse to fine realisations of the original image, (d-f) coarse to fine multiresolution segmentations, (g) single resolution segmentation*

Returning to the segmentation problem, having obtained GMRF parameter estimates at multi-resolutions, it is necessary to address the consequences of the resolution transform on the complete image model and in particular, on the MRF prior distribution $p(\mathbf{x} \mid \beta)$. Typically this takes the form of a Potts model for which no similar recursive relationship between realisations at different resolutions exists. Krishnamachari & Chellappa [63] suggest choosing empirically specific values for the β parameters at individual resolutions. They justify this by stating, "*fortunately, segmentation results are not heavily dependent on this parameter.*" Although this most certainly is unlikely to be the case, hence the initial requirement for differing hyper-parameter values at each resolution, experimental evidence suggests there is some degree of transience to the β parameter value required to segment similar types of image.

The results of applying such an algorithm are shown in Figure 2.6. The texture models used were 24 parameter GMRF's and parameter estimates were obtained *a priori*, at the finest resolution. The image was subsampled twice to produce two coarser realisations and GMRF parameters were obtained using the above methodology. The described segmentation algorithm was used with β parameters from the finest to coarsest of resolutions taking values 1.6, 1.2, 0.8, respectively. For comparison, a segmentation of the same image using a simulated annealing algorithm at a single resolution is shown, generated over an identical number of iterations to the multiresolution example.

2.5 Semi-unsupervised Segmentation

If each region within an image is modelled by an individual distribution then at each site one may assign a state which prescribes the particular distribution used to model the region of which the site is a member. The observed image may be treated as a joint realisation from a spatial process prescribing the juxtapositioning of these states on the image lattice and also realisations from each of the regional model distributions assigned to each site. Image segmentation can therefore be treated as an incomplete data problem [92] in which: the intensity data $\mathbf{y}$ is observed; the state data $\mathbf{x}$ is missing or hidden; the set of model parameters associated with each class $\boldsymbol{\theta}^{(Y)} = \{\boldsymbol{\theta}_c^{(Y)}, c \in \boldsymbol{\Lambda}\}$, need to be estimated, as do those parameters describing the distribution of labels on the underlying map, specifically $\boldsymbol{\theta}^{(X)}$.

2.5.1 Maximum *a posteriori* approaches

Using the notation just outlined, the problem may be posed as an optimisation problem:

$$\begin{aligned} &\arg \max_{\mathbf{x}\in\mathbf{X},\ \boldsymbol{\theta}^{(Y)}\in\boldsymbol{\Theta}^{(Y)},\ \boldsymbol{\theta}^{(X)}\in\boldsymbol{\Theta}^{(X)}} p\big(\mathbf{x}, \boldsymbol{\theta}^{(Y)}, \boldsymbol{\theta}^{(X)} \mid \mathbf{y}\big) \\ &= \arg \max_{\mathbf{x}\in\mathbf{X},\ \boldsymbol{\theta}^{(Y)}\in\boldsymbol{\Theta}^{(Y)},\ \boldsymbol{\theta}^{(X)}\in\boldsymbol{\Theta}^{(X)}} p\big(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^{(Y)}\big)\ p\big(\mathbf{x} \mid \boldsymbol{\theta}^{(X)}\big)\ p\big(\boldsymbol{\theta}^{(Y)}\big)\ p\big(\boldsymbol{\theta}^{(X)}\big) \end{aligned} \tag{2.52}$$

Due to the modelling of the underlying state process by a Markov Random Field the direct solution of the above optimisation problem is both analytically and computationally intractable. This is due to both the complexity of calculating the partition function on the prior $p(\mathbf{x} \mid \boldsymbol{\theta}^{(X)})$, and due to the typically multimodal nature of the posterior distribution.

To overcome these problems, an approach often adopted estimates a *partial optimal solution* [96]. The partial optimal solution $\{\mathbf{x}^*, \boldsymbol{\theta}^{(Y)^*}, \boldsymbol{\theta}^{(X)^*}\}$, is defined by the following equations:

$$\mathbf{x}^* = \arg \max_{\mathbf{x}\in\mathbf{X}} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^{(Y)^*})\ p\big(\mathbf{x} \mid \boldsymbol{\theta}^{(X)^*}\big)\ p\big(\boldsymbol{\theta}^{(Y)^*}\big)\ p\big(\boldsymbol{\theta}^{(X)^*}\big) \tag{2.53}$$

$$\boldsymbol{\theta}^{(Y)^*} = \arg \max_{\boldsymbol{\theta}^{(Y)}\in\boldsymbol{\Theta}^{(Y)}} p\big(\mathbf{y} \mid \mathbf{x}^*, \boldsymbol{\theta}^{(Y)}\big)\ p\big(\boldsymbol{\theta}^{(Y)}\big) \tag{2.54}$$

$$\boldsymbol{\theta}^{(X)^*} = \arg \max_{\boldsymbol{\theta}^{(X)}\in\boldsymbol{\Theta}^{(X)}} p\big(\mathbf{x}^* \mid \boldsymbol{\theta}^{(X)}\big)\ p\big(\boldsymbol{\theta}^{(X)}\big) \tag{2.55}$$

Superficially these equations appear identical to the criteria expressed in equation 2.52, however their incorporation into an optimisation algorithm results in a process that may well settle at a local minima in the objective function energy surface. Several approaches have been adopted using the partial optimal solution criteria. These will generally fall into one of two categories [96]; firstly, one in which the state configuration is optimised according to some form of relaxation algorithm which is interrupted at regular intervals, to update the maximum likelihood or *a posteriori* parameter estimates; or secondly, where each of the optimisations in equations 2.53 to 2.55 are completed alternately.

An example of the first of these methods, commonly known as *adaptive simulated annealing* (see Kato *et al*[60]) comprises a simulated annealing process for the optimisation of equation 2.53 while concurrently updating the model parameter maximum *a posteriori* estimates, conditional on the current realisation of the underlying field $\mathbf{x}$. To prove convergence to a partially optimal solution Lakshmanan & Derin[64] extended Geman & Geman's theorem [39] proving convergence for simulated annealing to show convergence in an optimisation process with unknown model parameters. A temperature schedule $T(t)$, was derived so that the optimisation process was guaranteed to reach a partially optimal

solution in infinite time. The schedule was such that

$$\lim_{t\to\infty} T(t) \;=\; 0 \tag{2.56}$$

$$T(t) \;\geq\; \frac{N\Delta^*}{\log(t)} \tag{2.57}$$

where N is the size of the image lattice and Δ^* denotes the maximum difference in local conditional energy between values assigned to any two states given any surrounding neighbourhood configuration.

Won & Derin[96] adopt a slightly different approach: realising that the maximum *a posteriori* parameter estimates for each individual region model may often be expressed analytically (particularly if using a Gaussian spatial process and assuming reference or conjugate priors), the optimisation process of equations 2.53 to 2.55 may be simplified to

$$\mathbf{x}^* \;=\; \arg\max_{\mathbf{x}\in\mathbf{X}} \; p\big(\mathbf{y} \mid \mathbf{x}, \hat{\theta}^{(Y)}(\mathbf{y},\mathbf{x})\big)\; p\big(\mathbf{x} \mid \boldsymbol{\theta}^{(X)^*}\big)\; p\big(\hat{\theta}^{(Y)}(\mathbf{y},\mathbf{x})\big)\; p\big(\boldsymbol{\theta}^{(X)^*}\big) \tag{2.58}$$

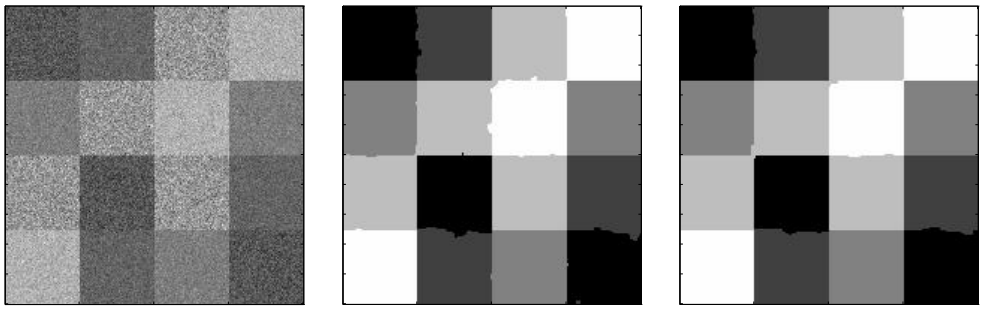
$$\boldsymbol{\theta}^{(X)^*} \;=\; \arg\max_{\boldsymbol{\theta}^{(X)}\in\boldsymbol{\Theta}^{(X)}} \; p\big(\mathbf{x}^* \mid \boldsymbol{\theta}^{(X)}\big)\; p\big(\boldsymbol{\theta}^{(X)}\big) \tag{2.59}$$

where $\hat{\theta}^{(Y)}(\mathbf{y},\mathbf{x})$ is the analytical maximum *a posteriori* parameter estimate. Having simplified the process to two steps, an optimisation algorithm is then adopted that alternately carries out the maximisation associated with equation 2.58 and then that associated with equation 2.59. To achieve the first of these steps, a simulated or a deterministic annealing process will be required. However, the second step often requires a less complex numerical optimisation process, depending on the convexity of the surface. Typically gradient descent methods suffice.

The estimation of hyper-parameters on the prior $p(\mathbf{x} \mid \boldsymbol{\theta}^{(X)})$ is a complex issue. This is due to the necessity of calculating the posterior probability to achieve the optimisation given in equation 2.59, which requires the evaluation of the Markov Random Field's partition function. Since this is computationally unfeasible, the pseudo-likelihood function is often substituted [11] [96] [64]. The pseudo-likelihood is simply the product of all conditional distributions on the Markov Random Field, given the current realisation, i.e.

$$\mathrm{PL}(\mathbf{x} \mid \boldsymbol{\theta}^{(X)}) \;=\; \prod_{s\in\boldsymbol{\Omega}} \frac{\exp\{-U(x_s \mid \boldsymbol{\theta}^{(X)}, \mathbf{x}_{\eta_s})\}}{\sum_{x\in\boldsymbol{\Lambda}} \exp\{-U(x \mid \boldsymbol{\theta}^{(X)}, \mathbf{x}_{\eta_s})\}} \tag{2.60}$$

where $U(\cdot \mid \cdot)$ is the local conditional energy function, $\boldsymbol{\Omega}$ is the set of points comprising the image lattice, η_s denotes the neighbourhood structure at site s, and $\boldsymbol{\Lambda}$ is the set of all possible states. An inherent problem with this approximation, which has to date been



(a) (b) (c)

Figure 2.7: *(a) original image, (b) estimate conditional on $\beta = 1.2$, (c) estimate conditional on $\beta = 1.861$*

neglected, is its tendency to break down under specific field configurations. This can be easily observed if one assumes a nearest neighbour structure when considering an Ising model for the prior distribution in the hierarchical model. The conditional energy function is given by $U(x_s \mid \boldsymbol{\theta}^{(X)}, \mathbf{x}_{\eta_s}) = \beta \sum_{r \in \eta_s} V(x_s \mid x_r)$. where the potential function $V(\cdot \mid \cdot)$ takes values from $\{-1, +1\}$ depending on whether the two labels comprising the second order clique, i.e. x_s and x_r, are assigned to identical or differing states, respectively.

Since there can only be four relevant neighbourhood configurations with respect to x_s (specifically, where either the state of zero, one, two, three or all four of the neighbouring pixels differs from that of x_s) the log prior on the underlying field will be given by

$$\begin{aligned} \log p(\mathbf{x} \mid \boldsymbol{\theta}_{\boldsymbol{X}}) \;=\; & 4\beta n_0 \;+\; 2\beta n_1 + n_2 \log \frac{1}{2} \;-\; 2\beta n_3 \;-\; 4\beta n_4 \\ & -\; (n_0 + n_4) \log \left[e^{4\beta} + e^{-4\beta} \right] \;-\; (n_1 + n_3) \log \left[e^{2\beta} + e^{-2\beta} \right] \end{aligned} \tag{2.61}$$

where n_0, n_1, n_2, n_3 and n_4 reflect the frequency of occurrence of each neighbourhood configuration within the current label field configuration. If one considers the case where β becomes very large, then the log prior probability will either tend to 0 or $-\infty$, depending on the existence of various configurations. More specifically, as $\beta \to \infty$,

$$\log p(\mathbf{x} \mid \boldsymbol{\theta}^{(\boldsymbol{X})}) \;\to\; n_2 \log \left(\frac{1}{2}\right) - 4n_3\beta \;-\; 8n_4\beta \tag{2.62}$$

This is illustrated in Figure 2.7 showing the segmentation of a Isotropic Gaussian Markov Random Field. The results presented use the methodology adopted by Won & Derin[96], described previously. Simulated annealing was used to optimise $\mathbf{x}$ and β was estimated using gradient descent methods. β was initially set to a value of 1.2 and 2.7(b) shows

the resulting optimisation for $\mathbf{x}$. From this β was recalculated as 1.861 and a further optimisation was obtained, shown in Figure 2.7(c). This configuration contains nearest neighbourhoods where at least two pixels are identical to that at the centre of the neighbourhood. Thus, the next estimate for β is unreliable since the posterior distribution for β has become improper: in fact $\beta \to \infty$.

Kato, Zerubia & Berthod [59] proposed several alternative methodologies for Potts model hyper-parameter estimation. They noted that given a model,

$$p(\mathbf{x}) \;=\; \frac{1}{Z(\beta)} \exp\Big\{ \sum_{s,r \in \eta_s} \delta(x_s, x_r) \Big\} \tag{2.63}$$

then by setting $\frac{\partial p(\mathbf{x})}{\partial \beta} = 0$, it is possible to obtain a maximum likelihood estimate (MLE) for β by solving

$$\begin{aligned} N^{ih}(\hat{\mathbf{x}}) \;&=\; \frac{\sum_{\mathbf{x}\in\mathbf{X}} N^{ih}(\mathbf{x}) \exp\{-\beta N^{ih}(\mathbf{x})\}}{\sum_{\mathbf{x}\in\mathbf{X}} \exp\{-\beta N^{ih}(\mathbf{x})\}} \\ &=\; \frac{\partial}{\partial \beta} Z(\beta) \end{aligned} \tag{2.64}$$

where $N^{ih}(\hat{\mathbf{x}})$ is the number of inhomogeneous neighbouring pairs or cliques in the MLE of the image segmentation and $\mathbf{X}$ is the complete set of possible image segmentations. The solution to this system is tractable because $\log Z(\beta)$ is convex in β and the gradient function may be approximated using stochastic relaxation [36]. Given the above equation it becomes possible to either estimate the hyper-parameter *a priori*, or to incorporate a hyper-parameter estimation step into an iterative segmentation algorithm: for example the Iterative Conditional Expectation (ICE) algorithm[3] is applied in [59] and [60]. However,

[3]The ICE algorithm is an approximate methodology which attempts to find parameter estimates in the absence of specific data, e.g. $\mathbf{x}$ in the hidden data problem. Given an estimator $\varepsilon_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{x})$, the ICE algorithm finds its expected value with respect to the unknown data distribution, i.e. $\hat{\boldsymbol{\theta}} = E_{\mathbf{x}}[\varepsilon_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{x})]$. Since the distribution for $\mathbf{x}$ is conditional on $\hat{\boldsymbol{\theta}}$, an iterative algorithm is defined by using the current parameter estimate $\hat{\boldsymbol{\theta}}$:

1. Set k=0 and choose an initial parameter estimate $\hat{\boldsymbol{\theta}}^k$,
2. Sample n realisations $\hat{\mathbf{x}}^i, i = 1, .., n$, from $p(\mathbf{x} \mid \hat{\boldsymbol{\theta}}^k, \mathbf{y})$,
3. Obtain $\hat{\boldsymbol{\theta}}^{k+1}$ by evaluating

$$\hat{\boldsymbol{\theta}}^{k+1} \;=\; E_{\mathbf{x}}[\varepsilon_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{x})] \;\approx\; \frac{1}{n} \sum_{i=1}^{n} \varepsilon_{\boldsymbol{\theta}}(\mathbf{y}, \hat{\mathbf{x}}^i)$$

4. If $\|\hat{\boldsymbol{\theta}}^{k+1} - \hat{\boldsymbol{\theta}}^k\| > \xi$ where ξ is an arbitrary precision specifier for the estimator $\varepsilon_{\boldsymbol{\theta}}$, goto 2, else stop.

either process is likely to be computationally intensive since the estimation of $N^{ih}(\mathbf{x})$ is of comparable difficulty to evaluating the partition function of the process.

2.5.2 Expectation Maximisation Algorithms

To perform semi-unsupervised (i.e. the number of classes is assumed to be known *a priori*) segmentation of image data, a method of concurrently estimating the underlying image and any associated model parameters is required. Alternatively, the problem may be viewed as one of parameter estimation from incomplete data. Following this approach, the complete data comprises $\mathbf{z} = \{\mathbf{y}, \mathbf{x}\}$, where $\mathbf{y}$ is observed and $\mathbf{x}$ is the underlying or hidden component. Applying this to image segmentation, $\mathbf{y}$ comprises the observed noisy or textured image and $\mathbf{x}$ is a lattice on which the segmentation of the image is defined. If any prior knowledge pertaining to the hidden data and the relationship between observed and hidden data is modelled by a distribution parameterised by the vector $\boldsymbol{\theta}$, i.e. $p(\mathbf{z} \mid \boldsymbol{\theta})$, then the parameter estimation problem is one of estimation from incomplete data.

The Expectation-Maximisation (EM) algorithm was first proposed by Dempster, Laird & Rubin[30] as an iterative maximum-likelihood procedure for parameter estimation from incomplete data. The methodology has been extensively applied to the problem of image segmentation [18] [66] [102] [47]. Since the EM algorithm yields maximum-likelihood parameter estimates but not necessarily a maximum-likelihood estimate for the hidden data, the semi-unsupervised segmentation problem is often solved using the following algorithm.

---

1. Obtain an initial parameter estimate $\boldsymbol{\theta}_0$ using an *ad hoc* procedure or an initial guess.
2. Use the EM algorithm to find the Maximum-Likelihood parameter estimate $\hat{\boldsymbol{\theta}}$.
3. Use a supervised algorithm to obtain the maximum-likelihood or maximum *a posteriori* estimate for the hidden data.

The EM algorithm is an iterative process where the k'th iteration consists of two steps. The first of these finds an expression for the expected value of the log likelihood over the hidden data $\mathbf{x}$, given the previous parameter estimate. The second step maximises this expectation over the parameter space. More formally, if k denotes the current iteration, the algorithm may be expressed

E-Step: Find the function $Q(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(k)}) = \mathrm{E}\big[\log p(\mathbf{x} \mid \boldsymbol{\theta}) \mid \mathbf{y}, \hat{\boldsymbol{\theta}}^{(k)}\big]$

M-Step: Find $\hat{\boldsymbol{\theta}}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(k)})$

When considering an MRF model the calculation of the expected value of the log-likelihood function is computationally challenging. There are two commonly used methodologies: to use Monte Carlo methods to sample from the distribution, or alternatively to use a soft decision approach. Chalmond [18] adopted the first approach: Gibbs sampling was employed to generate realisations of the underlying Markov Random Field (or hidden data) as if drawn from the distribution $p(\mathbf{x} \mid \mathbf{y}, \hat{\boldsymbol{\theta}}^{(k)})$; statistics were then generated from these realisations to facilitate the required maximisation of the objective functions comprising the second step. This method is computationally extremely inefficient since at each iteration of the algorithm multiple samples must be drawn from the Markov Random Field's joint distribution.

Zhang *et al* [102] proposed a method based upon re-writing the model using soft decisions for the underlying data. To achieve this the conditional Gibbs energy was redefined, replacing the hidden label at site s, namely x_s, with an indicator vector $\mathbf{x}_s$. Under a hard decision criterion this indicator vector takes its value from a set of unit vectors $\{\mathbf{e}_0, \mathbf{e}_1, ..., \mathbf{e}_K\}$ whose unitary element designates the state assigned to site s.

To incorporate a soft decision framework into the algorithm the posterior distribution must first be redefined to use indicator vectors. For ease of understanding, the case where spatial independence is assumed initially for both the likelihood $\mathbf{U}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^{(Y)})$ and prior $\mathbf{V}(\mathbf{x} \mid \boldsymbol{\theta}^{(X)})$ components of the energy function is first considered. The log posterior distribution can be written

$$\log p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}) = \sum_{s \in \Omega} \Big\{ \mathbf{x}_s^T \, \mathbf{U}_s(y_s \mid \boldsymbol{\theta}^{(Y)}) + \mathbf{x}_s^T \mathbf{V}_s(\boldsymbol{\theta}^{(X)}) \Big\} - \log Z(\mathbf{y}, \boldsymbol{\theta}) \tag{2.65}$$

If $\hat{\boldsymbol{\theta}}^{(k)}$ is the previous parameter vector estimate, then taking the expectation of the above equation over the state space for $\mathbf{x}$, i.e. $\mathbf{X}$, yields the appropriate function for maximisation in the second step of the EM algorithm,

$$Q(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(k)}) = \sum_{s \in \Omega} \mathrm{E}\big[\mathbf{x}_s^T \mid y_s, \hat{\boldsymbol{\theta}}^{(k)}\big] \, \big\{ \mathbf{U}_s(y_s \mid \boldsymbol{\theta}^{(Y)}) + \mathbf{V}_s(\boldsymbol{\theta}^{(X)}) \big\} - \log Z(\mathbf{y}, \boldsymbol{\theta}) \tag{2.66}$$

Here the i'th elements of the two vectors $\mathbf{U}_s(y_s \mid \boldsymbol{\theta}^{(Y)})$ and $\mathbf{V}_s(\boldsymbol{\theta}^{(X)})$ are given by

$$\mathbf{U}_s^{(i)}(y_s \mid \boldsymbol{\theta}^{(Y)}) = \log p(y_s \mid \mathbf{x}_s = \mathbf{e}_i, \boldsymbol{\theta}^{(Y)}) \tag{2.67}$$

$$\mathbf{V}_s^{(i)}(\boldsymbol{\theta}^{(X)}) = \log p(\mathbf{x}_s = \mathbf{e}_i \mid \boldsymbol{\theta}^{(X)}) \tag{2.68}$$

Soft decisions may now be implemented by re-defining the indicator vector so that its i'th component at the $k+1$'th iteration, specifically $x_s^{(i)^{(k+1)}}$, is equal to the i'th component of the conditional expectation;

$$x_s^{(i)^{(k+1)}} \triangleq \frac{p\left(\mathbf{x}_s = \mathbf{e}_i, \mid y_s, \hat{\boldsymbol{\theta}}^{(k)}\right)}{\sum_{j\in\Lambda} p\left(\mathbf{x}_s = \mathbf{e}_j, \mid y_s, \hat{\boldsymbol{\theta}}^{(k)}\right)} \tag{2.69}$$

Thus,

$$\begin{aligned} \mathrm{E}\left[\mathbf{x}_s \mid y_s, \hat{\boldsymbol{\theta}}^{(k)}\right] &= \sum_{i\in\boldsymbol{\Lambda}} \frac{p\left(\mathbf{x}_s = \mathbf{e}_i, \mid y_s, \hat{\boldsymbol{\theta}}^{(k)}\right)}{\sum_{j\in\Lambda} p\left(\mathbf{x}_s = \mathbf{e}_j, \mid y_s, \hat{\boldsymbol{\theta}}^{(k)}\right)} \mathbf{e}_i \\ &= \mathbf{x}_s^{(k+1)} \end{aligned} \tag{2.70}$$

An obvious methodology now exists by which the quantity given by equation 2.66 may be calculated, simply set all elements of the $\mathrm{E}\left[\mathbf{x}_s \mid y_s, \hat{\boldsymbol{\theta}}^{(k)}\right]$ to the normalised probability $p(\mathbf{x}_s = \mathbf{e}_i, \mid y_s, \hat{\boldsymbol{\theta}}^{(k)})$. The expected value of the log-likelihood function may then be calculated directly using these probabilities. If an independent Gaussian distribution is assumed for the likelihood function then the M-Step becomes trivial via simple calculus (relevant formulae are given in [102]).

When considering a Markov Random Field prior on $\mathbf{x} \in \mathbf{X}$ the above methodology breaks down. For example, consider the prior distribution on a Markov Random Field with support over the nearest neighbourhood $\eta_s = \{s+\tau_1, s+\tau_2, s+\tau_3, s+\tau_4\}$, with clique potentials $-\beta$ and $+\beta$. The conditional distribution is defined by the equation,

$$p\left(\mathbf{x}_s = \mathbf{e}_k \mid \{\mathbf{x}_r, r \in \eta_s\}, \boldsymbol{\theta}^{(x)}\right) \propto \exp\left\{\beta[2\mathbf{e}_k - 1]^t \left[\mathbf{x}_{s+\tau_1}, \mathbf{x}_{s+\tau_2}, \ ..., \mathbf{x}_{s+\tau_n}\right]\right\} \tag{2.71}$$

The expression given by equation 2.69 no longer applies due to an implied conditioning on neighbouring elements of $\mathbf{x}_s$. To overcome this problem Zhang *et al* [102] proposed the use of the Pseudo-likelihood function [11] to approximate the partition function, while conditioning on the previous estimate for $\mathbf{x}$. The resulting expression that replaces equation 2.69 is therefore

$$x_s^{(i)^{(k+1)}} \triangleq \mathrm{E}\left[x_s^{(i)} \mid y_s, \{\mathbf{x}_r^{(k)}, r \in \eta_s\}, \hat{\boldsymbol{\theta}}^{(k)}\right] = \frac{p\left(\mathbf{x}_s = \mathbf{e}_i, \mid y_s, \{\mathbf{x}_r^{(k)}, r \in \eta_s\}, \hat{\boldsymbol{\theta}}^{(k)}\right)}{\sum_{j\in\Lambda} p\left(\mathbf{x}_s = \mathbf{e}_j, \mid y_s, \{\mathbf{x}_r^{(k)}, r \in \eta_s\}, \hat{\boldsymbol{\theta}}^{(k)}\right)} \tag{2.72}$$

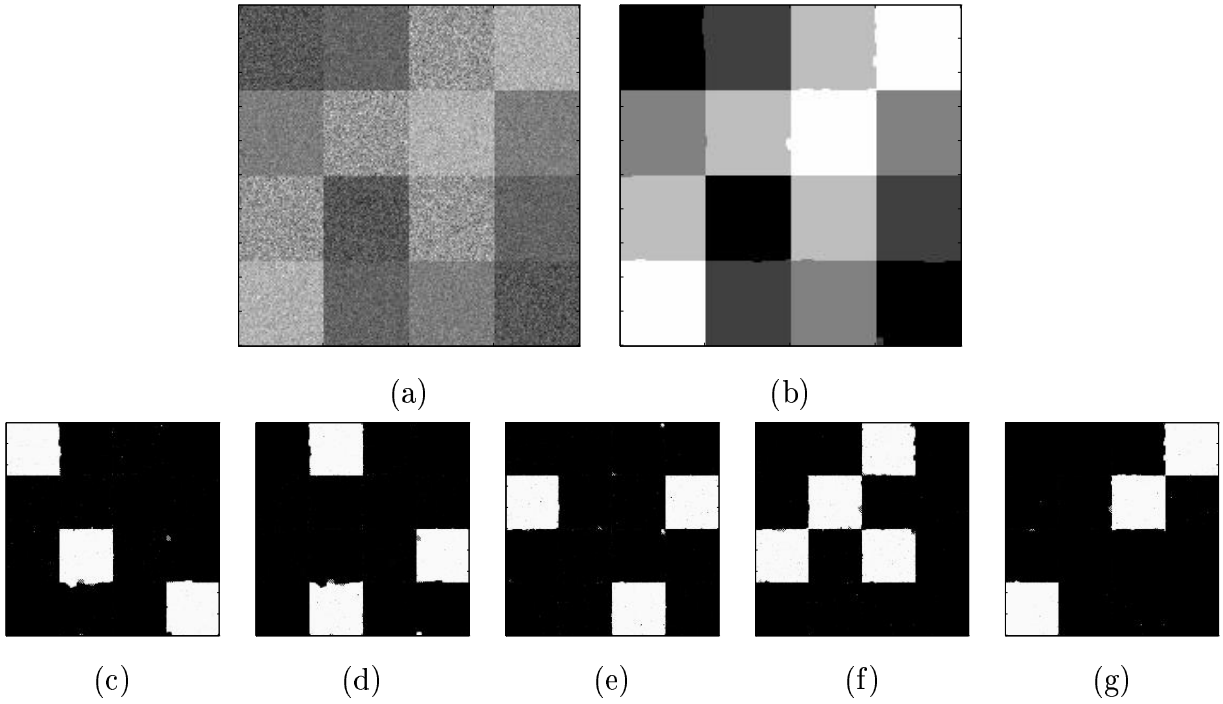


Figure 2.8: *(a) original image, (b) final segmentation, (c)-(g) the five final state probability planes*

The expected log-likelihood function is now approximated by

$$Q(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(k)}) = \sum_{s\in\Omega} \mathrm{E}\Big[\mathbf{x}_s^T \mid y_s, \hat{\boldsymbol{\theta}}^{(k)}, \{\mathbf{x}_r^{(k)}, r \in \eta_s\}\Big] \Big[\mathbf{U}_s(y_s \mid \boldsymbol{\theta}^{(Y)}) + \mathbf{V}_s\big(\{\mathbf{x}_r^{(k)}, r \in \eta_s\}, \boldsymbol{\theta}^{(X)}\big)\Big] - \sum_{s\in\Omega} \log \sum_{i\in\Lambda} \exp\Big\{\mathbf{e}_i \mathbf{V}_s\big(\{\mathbf{x}_r^{(k)}, r \in \eta_s\}, \boldsymbol{\theta}^{(X)}\big)\Big\} \tag{2.73}$$

For Isotropic and Gaussian cases the maximisation of this function is usually analytical with respect to the likelihood function parameters. If not, various approximations based on least squares estimates[68] or the Pseudo-likelihood may be used. Numerical procedures based on the Pseudo-likelihood approximation will generally be required to find the estimates of the Gibbs prior parameters. Results for the Isotropic Gaussian model are shown in Figure 2.8. The five state planes are shown in Figures 2.8(c)-(g), each of which comprise the elements of the final soft decision vectors across the whole image. These are found to be poorer estimates of the original segmentation than that obtained by simulated annealing via the Geman & Geman[39] approach, using the soft EM algorithm's estimated parameters (see Figure 2.8(b)).

Interestingly, if the soft decision vectors are replaced by hard decisions then the resulting algorithm would be identical to the Iterative Conditional Mode (ICM) algorithm of Besag [11]. In fact the soft decision algorithm is related to the Mean Field approaches adopted from statistical mechanics which will be discussed in the following section (section 2.5.3). Convergence is generally found to be better for soft decision algorithms as opposed their ICM counterparts [102], however, as with all EM algorithms, convergence is only guaranteed to a local minimum in the energy surface.

2.5.3 Mean Field Theory and Techniques

If $\mathbf{x}$ is an MRF then its joint probability may be expressed as a Gibbs distribution $p(\mathbf{x}) = Z^{-1}\exp\{-\beta U(\mathbf{x})\}$, where Z is the partition function and β represents inverse temperature. The energy function can be written in terms of clique potential functions $U(\mathbf{x}) = \sum_c V_c(\mathbf{x})$. This leads to a mean value at site i given by

$$\langle x_i \rangle \;=\; \frac{1}{Z}\sum_{\mathbf{x}\in\mathbf{X}} x_i \exp\left\{ -\beta\left[\sum_i V_c(x_i) + \frac{1}{2}\sum_{j\in N_i} V_c(x_i, x_j)\right]\right\} \tag{2.74}$$

where N_i is the first-order neighbourhood of site i.

Mean field theory [19] concerns the evaluation of the mean field $\langle \mathbf{x} \rangle$ over this distribution. Justification for the use of Mean Field estimates in optimisation problems stems from the fact that the Mean Field represents the minimum variance Bayes estimator for $\mathbf{x}$. This follows if the variance of field $\mathbf{x}$ from its centre $\hat{\mathbf{x}}$ is expressed by

$$\mathrm{var}_{\hat{\mathbf{x}}} \;=\; \textstyle\sum_{\mathbf{x}}(\mathbf{x}-\hat{\mathbf{x}})^2 p(\mathbf{x}) \quad \text{then} \quad \frac{\partial}{\partial \hat{\mathbf{x}}}\mathrm{var}_{\hat{\mathbf{x}}} \;=\; 0 \;\Rightarrow\; \hat{\mathbf{x}} = \textstyle\sum_{\mathbf{x}} \mathbf{x} p(\mathbf{x}) \tag{2.75}$$

The mean field assumption maintains that the interactions between the spins (or states) at all lattice sites and one particular site can be approximated by the force of a mean field on the particle at that site. This results in an approximation to the energy function which is site specific and can be divided into two components, one dependent, the other independent of site i, i.e.

$$U_i^{mf}(\mathbf{x}) \;=\; U(\mathbf{x})\,\big|_{x_j=\langle x_j\rangle} \tag{2.76}$$

$$\;=\; U_i^{mfle}(x_i) \;+\; U_i^{r}(\langle \mathbf{x}_{\mathbf{\Omega}\setminus i}\rangle) \tag{2.77}$$

where $U_i^{mfle}(x_i)$ is the mean field local energy at site i and $U_i^{r}(\langle \mathbf{x}_{\mathbf{\Omega}\setminus i}\rangle)$ is the remainder containing all terms not involving x_i. Derived from this equation is the mean field partition

function,

$$Z_i^{mf} = \sum_{x_i} \exp\{-\beta U_i^{mf}(\mathbf{x})\} \quad (2.78)$$

$$= \sum_{x_i} \exp\{-\beta[U_i^{mfle}(x_i) + U_i^r(\langle \mathbf{x}_{\mathbf{\Omega}\setminus i}\rangle)]\} \quad (2.79)$$

$$= \exp\{-\beta U_i^r(\langle \mathbf{x}_{\mathbf{\Omega}\setminus i}\rangle)\} \sum_{x_i} \exp\{-\beta U_i^{mfle}(x_i)\} \quad (2.80)$$

$$= Z_i^{mfle} \exp\{-\beta U_i^r(\langle \mathbf{x}_{\mathbf{\Omega}\setminus i}\rangle)\} \quad (2.81)$$

It follows that the mean value at site i, given by equation 2.74, may be approximated:

$$\langle x_i\rangle \approx \frac{1}{Z_i^{mf}} \sum_{x_i} x_i \exp\{-\beta[U_i^{mfle}(x_i) + U_i^r(\langle \mathbf{x}_{\mathbf{\Omega}\setminus i}\rangle)]\} \quad (2.82)$$

$$= \frac{1}{Z_i^{mfle}} \sum_{x_i} x_i \exp\{-\beta U_i^{mfle}(x_i)\} \quad (2.83)$$

Hence the mean field assumption implies that the force on each site is only attributable to the mean fields of neighbouring sites. The ability to calculate the mean field component at site i in terms of its neighbours allows the complete mean field to be found using an iterative update procedure. If the inverse temperature parameter β is increased so that $\beta \to \infty$, then the system will undergo an annealing process, converging to the MAP estimate. This is known as Mean Field Annealing (MFA).

Geiger & Girosi[34] proposed a further approximation to the Gibbs energy function which enables an estimate of the partition function to be generated. When considering the potential at each site, fluctuations at neighbouring sites are ignored, hence the energy function may be approximated,

$$U(\mathbf{x}) = \sum_i \left(V_c(x_i) + \frac{1}{2} \sum_{j\in N_i} V_c(x_i, x_j) \right) \quad (2.84)$$

$$= \sum_i \left(V_c(x_i) + \frac{1}{2} \sum_{j\in N_i} V_c(x_i, \langle x_j\rangle) \right) \quad (2.85)$$

This leads to a difference in the equivalent expression for equation 2.83: the value of the second order clique potential functions becomes halved. The new mean field approximation effectively makes all variables independently distributed, hence its partition function consists of a product of conditional distributions,

$$Z^{mf} = \prod_i \sum_{x_i} \exp\left\{ -\sum_i \left(V_c(x_i) + \frac{1}{2} \sum_{j\in N_i} V_c(x_i, \langle x_j\rangle) \right) \right\} \quad (2.86)$$

The similarity between this expression and Besag's pseudo-likelihood function [11] is immediately apparent, the only difference being the conditioning on mean values rather than simply current states. However, the mean field assumption has an obvious and plausible physical interpretation, while the pseudo-likelihood is at best an *ad hoc* approximation. In fact it may be beneficial to view the mean field approximation as evidence for the structure of the pseudo-likelihood, particularly in annealing applications. Here, the mean field estimate tends towards that which would minimise the energy function as the system temperature approaches zero.

The Mean Field approximation to the partition function may be used to derive further site update equations of a simpler form to that of equation 2.83. This is achieved by expressing the the mean value at a site i in terms of the true partition function[34]. Then, by substituting the Mean Field approximation to the partition function or alternatives, e.g. the saddle point approximation[35], useful update equations or 'Mean Field Equations' may be derived. For example, consider a typical Gibbs energy function,

$$U(\mathbf{x}) = \sum_i \Big[(y_i - x_i)^2 + \sum_{j \in \eta_i} U(x_i, x_j)\Big] \tag{2.87}$$

where η_i defines a neighbourhood structure surrounding site i and y_i is the corresponding observed data. Then the mean $\hat{x}_i$ may be expressed in terms of a partial derivative of the partition function;

$$y_i - \hat{x}_i = \frac{1}{Z} \sum_{\mathbf{x}} (y_i - x_i) \exp\Big\{ -\beta \sum_i \Big[(y_i - x_i)^2 + \sum_{j \in \eta_i} U(x_i, x_j)\Big]\Big\} \tag{2.88}$$

$$= -\frac{1}{2\beta Z} \sum_{\mathbf{x}} \frac{\partial}{\partial y_i} \exp\Big\{ -\beta \sum_i \Big[(y_i - x_i)^2 + \sum_{j \in \eta_i} U(x_i, x_j)\Big]\Big\} \tag{2.89}$$

$$= -\frac{1}{2\beta Z} \frac{\partial Z}{\partial y_i} \tag{2.90}$$

thus,

$$\hat{x}_i = y_i + \frac{1}{2\beta} \frac{\partial \ln Z}{\partial y_i} \tag{2.91}$$

By substituting one of the Mean Field or saddle point approximations to the partition function into the above equation, deterministic site update equations, or mean field equations may be derived.

2.5.4 Mean Field Annealing applied to the Weak Membrane Model

Mean Field Annealing (MFA) is particularly applicable to the weak membrane model as the model's energy function is expressed in terms of continuous gray scale values. Geiger & Girosi[34] developed such an algorithm based on the mean field and saddle point approximations to the partition function.

The weak membrane model consists of three coupled MRF's, one of which imposes smoothness on the image, while the remaining pair constitute horizontal and vertical line processes that locally decouple the smoothness field. Thus by defining: a lattice $\mathbf{\Omega}$, indexed by the pair (i,j); the observed data $\mathbf{y}$; the smoothing MRF $\mathbf{x}$; and the two line processes $\mathbf{h}$ and $\mathbf{v}$; then the energy function of the weak membrane model may be written

$$U(\mathbf{x},\mathbf{h},\mathbf{v}) = \sum_{(i,j)\in\Omega} \frac{1}{2\sigma^2}(y_{i,j}-x_{i,j})^2 \; + \; \left[(x_{i,j}-x_{i,j+1})^2(1-v_{i,j}) + (x_{i,j}-x_{i+1,j})^2(1-h_{i,j})\right] \\ + \; \gamma^h_{i,j}h_{i,j} + \gamma^v_{i,j}v_{i,j} \tag{2.92}$$

where $\gamma^h_{i,j}$ and $\gamma^v_{i,j}$ give the prior probabilities of the line processes. Hence, the higher the value a line element takes from the real line segment [0,1], the greater the degree of decoupling in the smoothness field between neighbouring sites. Recalling that the above expression constitutes the energy function of a Gibbs distribution, Geiger & Girosi[34] showed that the Gibbs partition function, i.e. $Z = \sum_{\{\mathbf{x},\mathbf{h},\mathbf{v}\}} \exp\{U(\mathbf{x},\mathbf{h},\mathbf{v})\}$, may be simplified by analytically evaluating the summations over the two line processes, giving

$$Z \;=\; \sum_{\mathbf{x}} \left\{ \exp\left\{ -\beta \sum_{i,j} [\frac{(x_{i,j}-y_{i,j})^2}{2\sigma^2} + \gamma^h_{i,j} + \gamma^v_{i,j}]\right\} \right. \\ \left. \times \prod_{i,j} \left(1 + e^{-\beta[\lambda(x_{i,j}-x_{i,j+1})^2 - \gamma^h_{i,j}]}\right)\left(1 + e^{-\beta[\lambda(x_{i,j}-x_{i+1,j})^2 - \gamma^v_{i,j}]}\right) \right\} \tag{2.93}$$

Using a process similar to that described in the previous section, (see equations 2.87 to 2.91) the mean values of the model's line process parameters can be written in terms of derivatives of the original partition function,

$$\hat{h}_{i,j} \;=\; -\frac{1}{\beta}\frac{\partial \ln Z}{\partial \gamma^h_{i,j}}\,, \qquad \hat{v}_{i,j} \;=\; -\frac{1}{\beta}\frac{\partial \ln Z}{\partial \gamma^v_{i,j}} \tag{2.94}$$

By applying these formulae to the saddle point approximation (i.e. $\sum_x e^{-\beta U(x)} \approx k\; e^{-\beta U(\hat{x})}$, where k is a constant and $\hat{x}$ satisfies $\frac{\partial U(x)}{\partial x} = 0$ such that $\hat{x} = \min_{x\in X} U(x)$)

for the partition function, given by equation 2.93 then the two mean fields $\mathbf{h}$ and $\mathbf{v}$ can be calculated iteratively using closed form update expressions, comprising simple sigmoid functions of the mean field $\hat{\mathbf{x}}$,

$$\hat{h}_{i,j} = \frac{1}{1+e^{\beta[\gamma^h_{i,j}-\lambda(\hat{x}_{ij}-\hat{x}_{i,j+1})^2]}}, \quad \hat{v}_{i,j} = \frac{1}{1+e^{\beta[\gamma^v_{i,j}-\lambda(\hat{x}_{ij}-\hat{x}_{i+1,j})^2]}} \tag{2.95}$$

The validity of using such approximations may be justified via an alternative derivation, see Appendix A. Here the mean values for the line process elements are evaluated without using the saddle point approximation and are shown to be given by similar expressions,

$$\langle h_{i,j}\rangle = \left\langle \frac{1}{1+e^{\beta[\gamma^h_{i,j}-\lambda(x_{ij}-x_{i,j+1})^2]}} \right\rangle_{\mathbf{x}}, \quad \langle v_{i,j}\rangle = \left\langle \frac{1}{1+e^{\beta[\gamma^v_{i,j}-\lambda(x_{ij}-x_{i+1,j})^2]}} \right\rangle_{\mathbf{x}} \tag{2.96}$$

From these expressions it is apparent that the saddle point approximation is identical to applying the mean field approximation to the expressions for the true mean values. Hence, its use is justifiable.

If equation 2.93 is treated as the summation over a Gibbs distribution, then by maximising the energy function with respect to $\mathbf{x}$, a set of equations are obtained which may be used as the basis for an optimisation algorithm. More efficiently, if the mean field approximation to the partition function is used, then by equating a similar set of partial derivatives to zero, i.e. $\frac{\partial Z}{\partial x_{i,j}} = 0$, a further set of update equations may be obtained,

$$\langle x_{i,j}\rangle = \frac{y_{i,j}+\lambda\sigma^2[\langle x_{i-1,j}\rangle(1-v_{i-1,j})+\langle x_{i,j+1}\rangle(1-h_{i,j})+\langle x_{i+1,j}\rangle(1-v_{i,j})+\langle x_{i,j-1}\rangle(1-h_{i,j-1})]}{1+\lambda\sigma^2[(1-v_{i-1,j})+(1-h_{i,j})+(1-v_{i,j})+(1-h_{i,j-1})]} \tag{2.97}$$

The results of applying such an algorithm are shown as Figure 2.9. For this experiment, the line process hyper-parameters are both set identically: $\gamma^h = \gamma^v = \gamma$. Figure 2.9(b) shows the smoothed image corresponding to the MRF $\mathbf{x}$, while Figure 2.9(c) combines the horizontal and vertical line processes in a single image.

An extended Mean Field annealing process was applied to the optimisation of Compound GMRF's (CGMRF) by Zerubia & Chellappa[99]. The CGMRF[53] is similar to the Weak Membrane model (see section 2.2.3) but allows the additional modelling of correlations between neighbouring sites.

2.5.5 Mean Field Annealing applied to Image Segmentation

As described previously in section 2.5.2, the problem of image segmentation may be viewed as one of parameter estimation from incomplete data. Frequently the Expectation - Maximisation(EM) algorithm [30] has been applied to such problems (see section 2.5.2). As

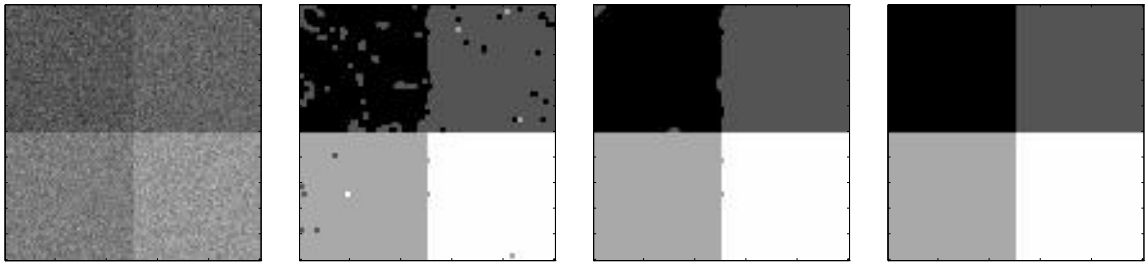
(a) (b) (c)

Figure 2.9: *Result of mean field annealing with an weak membrane model; (a) the original image, (b) the estimated underlying image and (c) the combined horizontal and vertical line processes. For this experiment* $\gamma = 200$ *and* $\lambda = 2$.

described previously, the EM algorithm is an iterative maximum-likelihood procedure to facilitate parameter estimation in incomplete data problems. Each iteration consists of two steps: The expectation, E-step and the maximisation, M-step. The E-step calculates the expectation of the likelihood of the hidden data given the observed data and current parameter estimates. The M-step then finds the parameter values that maximise this expectation.

However, the evaluation of expectation at each iteration comprises a major computational challenge, especially when considering a multivariate Gibbs distribution, such as an MRF joint distribution. The use of mean field procedures to calculate the expectation appears far more intuitive. Both the calculation of the Gibbs partition function and the calculation of the expected log likelihood are directly addressed. However several difficulties must be overcome. Throughout this chapter mean field optimisation processes have only been described where the MRF energy function is defined on a continuous state space. When considering segmentation or the hidden data problem, the state space or in this case the label space is discrete.

When considering a deterministic algorithm, a particular area of concern is the likelihood of settling in local minima. With mean field annealing on a discrete label space this is a particular problem; it will always be possible to find certain spatial configurations of pixels, that will remain stable when using local update equations such as 2.97, thus inhibiting the progression towards a lower energy realisation of the field. For example, in the case



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

Figure 2.10: *Original Image (a) and final segmentation after 200 iterations of mean field annealing using Potts models of varying support, (b) n=2, (c) n=3, (d) n=4.*

of a nearest neighbourhood MRF, the 4×4 pixel block forms a stable entity. However, by expanding the support of the MRF, the problem may be largely overcome, as demonstrated in figure 2.10, where the mean is effectively calculated over a larger area of the image and thus likelihood of the the neighbourhood configuration probabilities balancing, recedes.

As described in section 2.5.2, Zhang[100] applied the Expectation - Maximisation(EM) algorithm to image segmentation by incorporating soft decision vectors into the model. Interestingly, because each soft state vector comprises elements that are indicative of the state allocation probabilities (see equation 2.69) then by equation 2.71 these are in fact mean field estimates. Hence, in this case the use of soft decision vectors and mean field estimation is synonymous.

A further interesting concept introduced by Zhang[100] but not fully explained, is the relationship between convergence and the annealing process. Traditionally, simulated annealing is defined on a full Gibbs distribution $p(\mathbf{x}) = \frac{1}{Z}\exp\{-\frac{1}{T}U(\mathbf{x})\}$, where Z is the partition function and $U(\mathbf{x})$, the free energy associated with the Markov process $\mathbf{x}$. Simulated annealing operates by sampling $\mathbf{x}$ from the Gibbs distribution while concurrently lowering the system temperature. Convergence for a particular temperature schedule, was proved in the limit when the number of iterations $\rightarrow \infty$, by Geman & Geman[39] (see section 2.3).

Zhang[100] proposes an alternative algorithm where the Gibbs distribution is factorised into likelihood and prior terms and annealing is carried out solely on the prior. Hence denoting the observed data $\mathbf{y}$, then the complete distribution is given by $p(\mathbf{x} \mid \mathbf{y}) = \frac{1}{Z}\exp\{-U_L(\mathbf{y} \mid \mathbf{x}) - \frac{1}{T}U_P(\mathbf{x})\}$, where $U_L(\cdot)$ is the energy function corresponding to the likelihood term and $U_P(\mathbf{x})$ is that of the prior. Temperature can only be lowered to

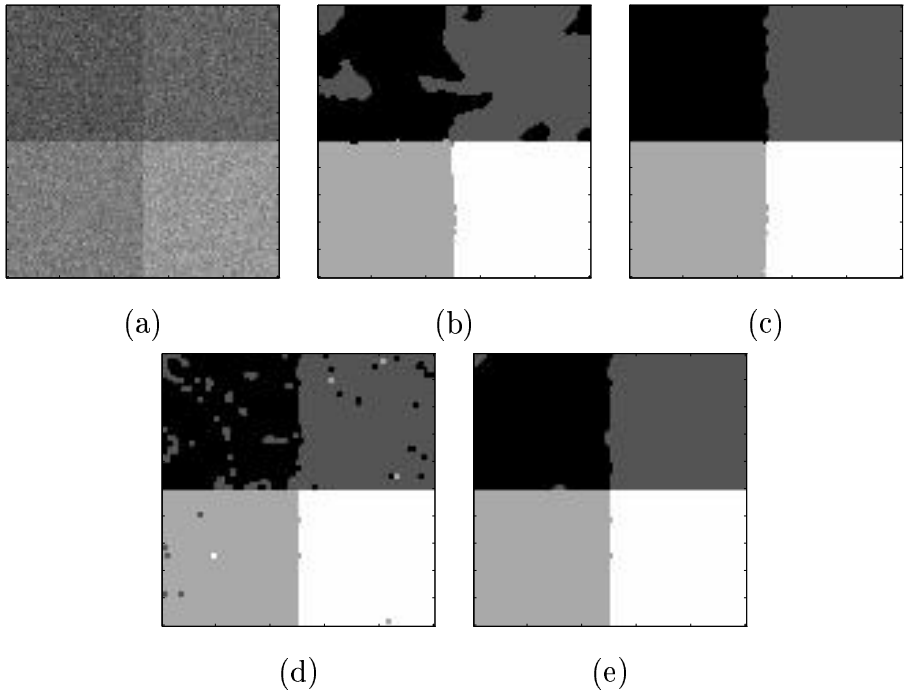


Figure 2.11: *Original Image (a) and final segmentations resulting from various types of annealing algorithm. (b) simulated annealing the full Gibbs distribution, (c) simulated annealing the prior only, (d) mean field annealing the full Gibbs distribution, (e) mean field annealing the prior only.*

unity and unfortunately no convergence proofs are available for such an algorithm but convergence experimental results (see Figure 2.11) appear to demonstrate an improvement in convergence for both mean field and simulated annealing algorithms.

2.6 Unsupervised Segmentation

Unsupervised segmentation is defined to be the segmentation of an image into an unknown number of homogeneous regions. The use of an hierarchical image model, defined in the introduction to this chapter, implies that unsupervised segmentation will consist of the concurrent estimation of the number of regions, the model parameters pertaining to each region, any Markov Random Field hyper-parameters and finally the actual segmentation.

The most obvious extension to the semi-supervised segmentation algorithms described

previously, was pioneered by Won & Derin [96] and Langan *et al* [66] to facilitate the estimation of the number of regions within the image. Using existing semi-supervised segmentation algorithms they proposed individually optimising a set of models, each of which was defined to model a specific number of regions. Then using some form of metric, often based on information criteria, the most '*probable*' model and consequent segmentation could be ascertained.

A second and thus far more widely accepted methodology, consists of various *two step* algorithms [68] [24] [76]. The first step is typically a coarse segmentation of the image to ascertain the most probable number of regions. This is achieved by dividing the image into windows, calculating features or estimating model parameters, then using a measure to combine or cluster closely related windows. The resulting segmentation is then used to estimate model parameters for each of the classes, before a supervised high-resolution segmentation is implemented using some form of relaxation algorithm. Since the estimation of regions is largely a feature based approach, these approaches will be discussed in more detail in the following chapter which is dedicated to statistical feature based segmentation methodologies.

To form an optimisation across the combined model space, a metric must exist to facilitate a comparison between models. The differing number of model parameters necessary to define models of different order, (in this case numbers of regions) means that it is non-informative to directly compare their maximum *a posteriori* probabilities, because each model distribution is defined over a different model space.

To overcome this difficulty various authors [96] [14] [66] have proposed using model likelihoods together with model fitting criteria. The concept was introduced to image processing by Zhang & Modestino [101] who suggested using a mixture model and clustering algorithm to ascertain the model order prior to segmentation. A criterion based on the Akaike Information Criteria (AIC) [1] [84] was used, given by

$$\mathrm{AIC}(k) \;=\; -\,2\log(\text{ML estimate for model } \;k) \;+\; 2n(k) \tag{2.98}$$

where k is the model order and $n(k)$ is the number of independent parameters associated with that model. The principle underpinning AIC is that of minimisation of the Kullback-Liebler distance or cross-entropy between the true distribution and a predictive model [44]. Due to its use of maximum likelihood estimates in the determination of a predictive model, i.e. ignoring uncertainty in the model parameters, the AIC is non-Bayesian. The formulation of the AIC fails to incorporate any dependency on the sample size, thus the

AIC is found to be biased for problems of model order selection[55]; the probability of the selecting the true model does not tend to unity as the sample size approaches infinity.

Bouman & Liu [14] applied this criterion to the unsupervised texture segmentation problem using multiresolution techniques. A hierarchical MRF model was used with textures being modelled by autoregressive (AR) processes. The algorithm began by taking the subsampled image at the coarsest resolution and dividing it into blocks. These were then clustered into M different classes using a vector of sufficient statistics $\boldsymbol{\xi}_k$, to form the MLE for the AR parameter vector, denoted $\hat{\boldsymbol{\theta}}_k$. If W_k gives a weighting for the number of blocks assigned to class k and A_k is the set of these blocks then the posterior distribution for this allocation may be written

$$\begin{aligned} p(\mathbf{y}, A_1, ...A_M \mid \hat{\boldsymbol{\theta}}_M, \boldsymbol{\rho}) &= \prod_{k=1}^{M} p(\boldsymbol{\xi}_{A_k} \mid \hat{\boldsymbol{\theta}}_k)\ p(W_k) \\ &\propto \exp\{L(\boldsymbol{\xi}_{A_k} \mid \hat{\boldsymbol{\theta}}_k) + W_k \log \rho_k\} \end{aligned} \tag{2.99}$$

where, $L(\boldsymbol{\xi}_k \mid \hat{\boldsymbol{\theta}}_k)$ is the energy function pertaining to the likelihood function for cluster k, and ρ_k is the parameter associated with a multinomial prior distribution for the weight parameter W_k. However, the problem of estimating the number of classes or regions M, is still poorly defined, thus Bouman & Liu [14] suggest the modification of equation 2.99 to incorporate the AIC. This results in the model selection criterion

$$-2 \log p(\mathbf{y}, A_1, ...A_M \mid \hat{\boldsymbol{\theta}}, \boldsymbol{\rho}) \ + \ 2MN_{\boldsymbol{\theta}_M} \tag{2.100}$$

where $N_{\boldsymbol{\theta}_M}$ is the number of model parameters pertaining to the model order M. Using this expression the number of classes M could be selected and the MLE's of the AR parameter vectors could then be used as the basis for a multiresolution segmentation algorithm.

The AIC is a single member of a complete family of model fitting criteria which have the general form,

$$\log L(k) \ - \ a(N)\ m(k) \ - \ b(k, N) \tag{2.101}$$

where: $L(k)$ is the likelihood for model k; $a(N)$ is a function of the sample size (in this case the image size); $m(k)$ is the number of independent model parameters, and $b(k, N)$ is a joint function of the two. The AIC is formed by setting $a(N) = 1$ and $b(k, N) = 0$.

To overcome this problem various forms of information criteria have been proposed based on Bayesian premises. These take various forms: Schwartz's criterion [85] where $a(N) = \frac{1}{2} \log N$ and $b(k, N) = 0$; Kashyap's criterion [58] where $a(N) = \frac{1}{2} \log N$ and

$b(k,N) = \log|B(k,N)|$ and $B(k,N)$ is the Hessian of the k'th model likelihood evaluated at the MLE for the model parameters; finally, Rissanen's criterion [83], where $a(N) = \frac{1}{2}\log\left(\frac{N+2}{24}\right)$ and $b(k,N) = \log(k+1)$. Rissanen also suggested a further criterion, often referred to as minimum description length (MDL) and given by $a(N) = \frac{1}{2}\log N$ and $b(k,N) = 0$. The formulation of this criterion is also asymptotically unbiased and has intuitive appeal since it minimises the combined description length of both the model data and parameters.

Won & Derin [96] approach the model selection problem problem somewhat differently. Due to the complexity of the partition functions, model selection is achieved on the basis of relative pseudo-likelihoods. Experimentally they observed that the above criteria tend to overfit to the model and so an alternative penalty term was proposed giving a model selection criterion,

$$\log L(k) \; - \; N^c m(k) \tag{2.102}$$

where c is a preselected constant. They state that this criterion is empirical and is thus somewhat arbitrary, (most obviously due to the need to prespecify the constant, c) but observe that the criteria obeys the general form of equation 2.101. Won & Derin [96] also state that,"*although each of the above model criterion satisfies a specific objective, they appear arbitrary, especially in comparison with each other.*" Such difficulties provide the motivation for the development of algorithms that make a direct comparison between posterior probabilities, as opposed to the use of likelihood ratios and information criteria, thus allowing any prespecified bias on the model order to be expressed, naturally in a Bayesian framework in terms of a prior distribution.

One further approach to model selection was adopted by Langan *et al* [66]. The EM algorithm approach, described in section 2.5.2, was used to optimise a set of different order models, prior to a model selection process. The model selection criterion adopted was postulated on the basis of empirical evidence, specifically,"*the log-likelihood often exhibits a rising exponential behaviour as a function of the model order.*" Hence, the expected log-likelihood function of equation 2.66 may be modelled by the expression

$$Q(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}, k) \; \approx \; \gamma[1 - \exp\{-\alpha k\}] \tag{2.103}$$

where α and γ are constants which are selected to minimise the least squares distance between this model and the observed range of values across the model space. Given this expression, the model order is selected that allows the log-likelihood to reach 90% of its

rise, i.e.,

$$\hat{k} = \arg \min_{1 \leq k \leq k_{max}} \left| 0.1 - \frac{\exp\{-\alpha k\}}{\exp\{-\alpha k_{max}\}} \right| \tag{2.104}$$

This approach appears to be as *ad hoc* as the more usual information criterion methods. However, Langan *et al* [66] justify this methodology with experimental evidence, suggesting that such a criterion yields improved results over those of a more theoretical nature.

These comprise just a few of the various information criteria used throughout model selection literature, see [44]. Since the evaluation of their suitability for image segmentation algorithms has been largely based on experimental evidence, the choice of criteria must be presumed subjective for this application.

2.7 Concluding Remarks

The concepts of supervised, semi-supervised and fully unsupervised image segmentation have now been formally introduced. In section 2.1 the criteria pertaining to these different problems have been specified, allowing the issues to be formulated in terms of a fully Bayesian framework, thus casting the various image segmentation issues in terms of the specific optimisation problems.

The hierarchical model is composed of individual region models, each a Markov Random Field (MRF). These region models are selected at each pixel by a labels which together specify the complete segmentation. The interaction between labels and thus regions is modelled using a further MRF.

Many types of models have been used to model image regions and textures, however the Gaussian MRF, introduced in section 2.2.3 has proved the most popular among researchers and thus this model will be used extensively throughout the remainder of this dissertation. Ising and Potts' models, used to model region interaction have been described in section 2.2.1. These, when combined with line or edge processes provide a powerful tool to model image structure. However, their behaviour when varying system temperature, particularly in the context of a simulated annealing algorithm must be treated with care.

Existing algorithms developed to implement the supervised, semi-supervised and fully unsupervised image segmentation processes have been described in sections 2.3, 2.5 and

2.6, respectively. Many model optimisation and parameter estimation techniques have been applied throughout the literature to hierarchical MRF models: Gibbs sampling; the Iterative Conditional Modes (ICM) algorithm; multi-resolution techniques; adaptive simulated annealing; the Iterative Conditional Expectation (ICE) algorithm; the Expectation-Maximisation (EM) algorithm; soft-decision vectors, and finally, the theory of mean fields. Unfortunately none of these techniques has proved to be globally convergent due to a tendency for both deterministic and stochastic optimisation algorithms to become stuck in local energy minima. However, results have proved encouraging and if a mechanism can be established which would allow an optimisation process to escape such local minima, a robust segmentation process would be ascertainable. This is the main motivation for the research presented in chapters 4 and 5 of this dissertation.

Approaches to the problem of estimating the number of regions or *unsupervised segmentation* were outlined in the previous section. However these have relied on the use of various information criteria, the choice of which is somewhat arbitrary. An alternative philosophy would assume a non-informative prior for model order and then define a posterior distribution over the combined model space. Thus both model parameter uncertainty and any prior knowledge concerning model order could be encompassed naturally in the problem formulation. The establishment of a mechanism which would then allow the joint MAP estimation of model order, parameters and segmentation using such a model is a central theme to chapters 4 and 5.

Feature Based Segmentation 3

3.1 Introduction

One of the fundamental methodologies of texture segmentation was developed by Geman *et al*[37] and is based on the following premise;

"it is well known that human's perceive 'textural' boundaries between regions of approximately the same brightness because the regions themselves, although containing sharp intensity changes are perceived as 'homogeneous' based on other properties, namely those involving the spatial distribution of intensity values."

The philosophy underpins a large collection of existing texture segmentation algorithms, all of which generate a feature vector at each point in the image by evaluating spatial properties over a surrounding region or window of pixels. Much previous effort has been spent trying to ascertain the principle components of a, in some sense, optimal feature vector for texture classification.

Perhaps the most obvious source of features are those derived from a prespecified image model. However, there are several other sources of interesting feature: for example, local spatial statistics derived from cooccurrence matrices and also, features derived from spatial filtering.

An approach widely adopted throughout image processing literature involves the use of two-step segmentation algorithms. The first step of such an algorithm achieves a coarse segmentation through the use of unsupervised segmentation or clustering techniques, based upon the derived feature vectors. Parameter estimates can then be obtained from the coarse segmentation which can be used as the basis for the second step, a supervised segmentation procedure, often comprising of a stochastic relaxation algorithm of the type described in

the previous chapter, see section 2.3.

The three main types of features outlined above and their associated segmentation and clustering algorithms will be examined in more detail in the following sections, specifically: section 3.2 discusses the use of statistics derived from image models; local spatial statistics are described in section 3.3, and wavelet and in particular Gabor filtering approaches to texture segmentation are reviewed in section 3.4. Finally, some conclusions pertaining to the usefulness of the various features and clustering algorithms are drawn in section 3.5.

3.2 Model Based Statistics

As suggested in the chapter introduction, if a model is postulated for the image in question, then the most obvious feature vector will comprise the model parameter estimates themselves, since by definition, these will form a lossless feature set with respect to our prior knowledge. Throughout the literature, hierarchical GMRF's are commonly chosen as region models. Having selected a set of features, some form of classification process is still required to obtain the desired segmentation. As described in the introduction these often comprise two-step procedures. Such algorithms will now be examined in more detail.

Having chosen the GMRF model parameters as features, Manjunath & Chellappa [68] propose segmenting textured images using a two-step algorithm: their first step uses these features in a coarse segmentation process; the second step, takes further parameter estimates using as a basis the newly obtained coarse segmentation and achieves a fine-scale segmentation using a model based supervised segmentation algorithm of the type described in the previous chapter, section 2.3.

To achieve a coarse segmentation the image is initially divided into a number of non-overlapping windows, for each of which feature vectors are calculated. As described, each feature vector comprises estimates of the GMRF model parameters, specifically $\hat{\boldsymbol{\phi}} = [\hat{\mu}, \hat{\sigma}^2, \hat{\boldsymbol{\theta}}]$. Least squares parameter estimates are used, which although compromising the stability of the estimated model, allow the estimates to be obtained with minimal computation burden. Adopting the GMRF model of section 2.2.3 with $\mathcal{W}$ as the $N \times M$ window of pixels and defining $y'_s = y_s - \hat{\mu}, s \in \mathcal{W}$ as the observed data normalised for mean, then

the GMRF parameter estimates are given by

$$\mathbf{Q}_s = [y_{s+\tau_1}, y_{s-\tau_1}, y_{s+\tau_2}, y_{s-\tau_2}, ..., y_{s+\tau_n}, y_{s-\tau_n}]^T \tag{3.1}$$

$$\hat{\boldsymbol{\theta}} = \left[\sum_{s\in\mathcal{W}} \mathbf{Q}_s \mathbf{Q}_s^T\right]^{-1} \left[\sum_{s\in\mathcal{W}} \mathbf{Q}_s y_s\right] \tag{3.2}$$

$$\hat{\boldsymbol{\sigma}}^2 = \frac{1}{NM} \sum_{s\in\mathcal{W}} \left[y_s - \hat{\boldsymbol{\theta}}^T \mathbf{Q}_s\right]^2 \tag{3.3}$$

A normalised distance is defined as a difference measure between the two feature vectors for the windows i and j:

$$d(\hat{\boldsymbol{\phi}}_i, \hat{\boldsymbol{\phi}}_j) = \sum_k \frac{(\hat{\boldsymbol{\phi}}_i(k) - \hat{\boldsymbol{\phi}}_j(k))^2}{(\hat{\boldsymbol{\phi}}_i(k))^2 + (\hat{\boldsymbol{\phi}}_j(k))^2} \tag{3.4}$$

Various forms of clustering algorithm incorporating this measure can now be implemented to arrive at a coarse segmentation. For example, Manjunath & Chellappa [68] suggest finding the maximum distance between feature vectors in the image, here denoted d_{max}, then allocating pixels i and j to the same cluster if and only if $d(\hat{\boldsymbol{\phi}}_i, \hat{\boldsymbol{\phi}}_j) < \rho d_{max}$. Unfortunately this requires pre-specification of the clustering parameter ρ, to set the degree of separation required in the measure that would allow the generation of a fresh cluster.

Such an algorithm is effective when segmenting images composed of large regions with distinct clusters, see the simple experiment shown in Figure 3.1(a-d), however, if the clusters are poorly defined then the algorithm performs poorly, see Figure 3.1(e-f). Another problem with adopting such a methodology stems from the necessity of using windowing to estimate model parameters: more detailed images, consisting of numerous small regions will be more difficult to segment.

To improve clustering performance Nguyen & Cohen [76] adopt a similar but more sophisticated approach. They apply an unsupervised soft decision or *fuzzy* variant of the K-means algorithm. The Fuzzy C-means algorithm [31] is a soft clustering algorithm that attempts to partition the data $\{x_i, i = 1, 2, ..., N\}$ into C clusters on the basis of a set of feature vectors $\{f_i, i = 1, 2, ..., N\}$. The final partition is expressed as an $N \times C$ matrix, denoted $\mathbf{U}$, whose elements u_{ij} are indicative of the probability of the data x_j belonging

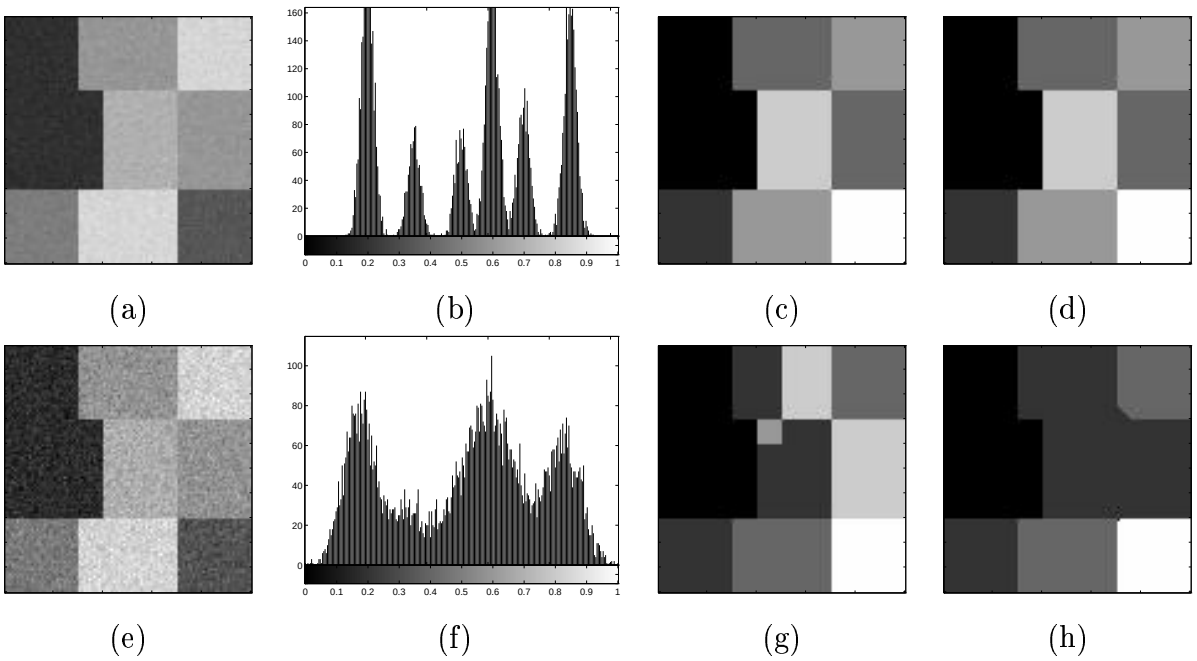


Figure 3.1: *Results of using Manjunath & Chellappa's [68] algorithm on a gray-scale image where each region may be parameterised by $\phi = [\mu, \sigma^2]$. Figures (a)&(e) give the original images, (b)&(f) the corresponding histograms, (c)&(g) the coarse segmentations and (d)&(h) the fine-scale segmentations.*

to cluster i. Hence u_{ij} enjoys the conditions

$$u_{ij} \in [0,1] \tag{3.5}$$

$$\sum_{j=1}^{N} u_{ij} > 0 \tag{3.6}$$

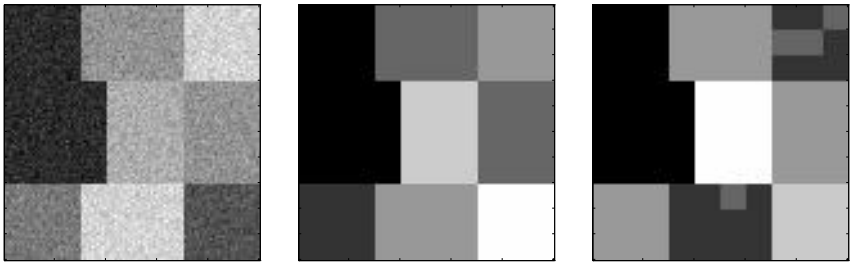
$$\sum_{i=1}^{C} u_{ij} = 1 \tag{3.7}$$

The C-means algorithm is configured to optimise the following objective function,

$$J_m(\mathbf{U}, \mathbf{V}) = \sum_{j=1}^{N} \sum_{i=1}^{C} u_{ij}^m d_{ij}^2 \tag{3.8}$$

where $\mathbf{V}$ is the set of cluster centre vectors $\mathbf{v}_i$, $d_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\|$ is generally the Euclidean distance in feature space and m is an arbitrary constant, pre-selected from $[1, \infty)$.

After selecting: m; C; the distance measure $\|\cdot\|$; choosing a stopping threshold ξ; then randomly initialising the matrix $\mathbf{U} \to \mathbf{U}^{(1)}$, the following iterative algorithm can be used



(a) (b) (c)

Figure 3.2: *Results of using Nguyen & Cohen's [76] clustering algorithm on a gray-scale image where each region may be parameterised by $\phi = [\mu, \sigma^2]$. Figure (a) is the original images, (b) the correct coarse segmentation, (c) the misclassified coarse segmentation.*

to carry out the required optimisation:

(1) set n=1

(2) calculate the $n'th$ set of cluster centres $\mathbf{V}^{(n)}$, given the matrix $\mathbf{U}^{(n)}$, according to
$v_i(k) = \frac{\sum_{j=1}^{N} u_{ij}^m f_j(k)}{\sum_{j=1}^{N} u_{ij}^m}$

(3) update $\mathbf{u}^{(n)} \rightarrow \mathbf{u}^{(n+1)}$ for each x_i by setting
$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}}$ if $\{i : d_{ij} = 0\} = \emptyset$,
$u_{ij} = 1$ for $j : d_{ij} = 0, u_{ij} = 0 \forall j : d_{ij} \neq 0$ if $\{i : d_{ij} = 0\} \neq \emptyset$,

(4) if $\|\mathbf{U}^{(n)} - \mathbf{U}^{(n+1)}\| < \xi$ then stop else set $n = n + 1$ and goto (2).

Here, the matrix norm in step (4) might take the form,
$\|\mathbf{U}^{(n)} - \mathbf{U}^{(n+1)}\| = \min_{i,j} |u_{ij}^{(n)} - u_{ij}^{(n+1)}|$.

Such an algorithm appears to exhibit improved convergence over the hard K-means version, however, there is no guarantee of convergence to the global mode: as with many optimisation algorithms there is a significant chance of reaching and remaining at a local minimum. A specific and common example occurs when data comprising a true C^* clusters converges to a $C^* - 1$ cluster realisation by effectively averaging means across two separate clusters, see Figure 3.2.

To extend the C-means algorithm to provide unsupervised clustering, a *plausibility measure* is imposed giving an indication of the quality of fit of data to the current C

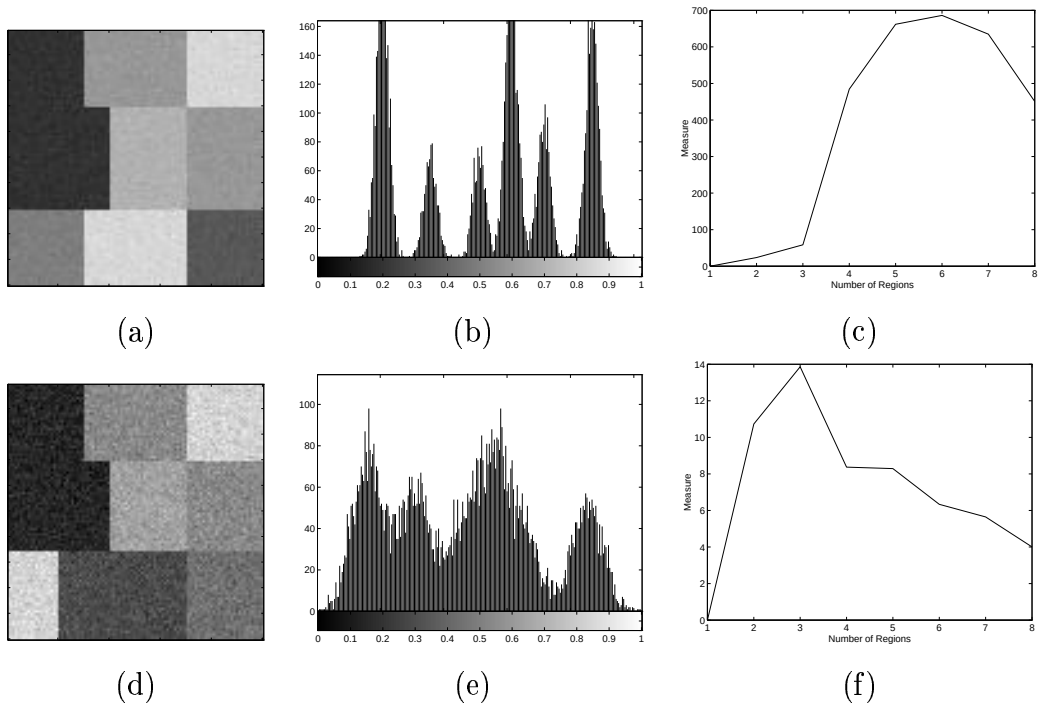


Figure 3.3: *Results of using Nguyen & Cohen's [76] unsupervised clustering algorithm on a gray-scale image. Figures (a,d) the original images, (b,e) the associated gray-scale histograms, (c,f) the corresponding values for measure given in equation 3.9 for 1-8 classes or regions.*

clusters. Adopting the above notation, the measure is given by

$$\mathcal{M}(\mathbf{U}, C) \;=\; \frac{\sum_{k=1}^{C} N_{\boldsymbol{\mathcal{C}}_k} \Gamma(k)}{\sum_{k=1}^{C} N_{\boldsymbol{\mathcal{C}}_k}} \tag{3.9}$$

where $N_{\boldsymbol{\mathcal{C}}_k}$ is the number of elements in cluster $\boldsymbol{\mathcal{C}}_k$ and

$$\Gamma(k) \;=\; \frac{\min_{j:j\neq k} d^2(\mathbf{v}_{\boldsymbol{\mathcal{C}}_k}, \mathbf{v}_{\boldsymbol{\mathcal{C}}_j})}{\frac{1}{N_{\boldsymbol{\mathcal{C}}_k}} \sum_{x \in \boldsymbol{\mathcal{C}}_k} d^2(\mathbf{v}_{\boldsymbol{\mathcal{C}}_k}, \mathbf{v}_{\boldsymbol{\mathcal{C}}_j})} \tag{3.10}$$

where $d(\mathbf{v}_{\boldsymbol{\mathcal{C}}_k}, \mathbf{v}_{\boldsymbol{\mathcal{C}}_j})$ is the Euclidean distance between the centres of clusters $\boldsymbol{\mathcal{C}}_k$ and $\boldsymbol{\mathcal{C}}_j$. The application of this criterion to the fuzzy C-means clustering algorithm is demonstrated in Figure 3.3. The measure is tested on two images with low (a-c) and higher noise (d-f) variances. The measure peeks as if fitting to a Gaussian mixture distribution. However, in the second case this represents a misclassification since the number of regions is under-estimated. The most obvious reason for this discrepancy is that despite using windowing to estimate model parameters, the spatial element to the data is under-utilised by the algorithm.

Panjwani & Healey [78] propose an algorithm which overcomes the problem. They propose a somewhat *ad hoc* clustering scheme that utilises the spatial relationship between windows. The algorithm comprises four steps: first the image is split into a set of non-overlapping square windows; next these are split iteratively until the data in each window obeys a homegeneity measure across it entirety, or else the window sized reached is the minimum allowable, in this case a 4×4 pixel area; next, a conservative merging algorithm is applied which allows like and also spatially adjacent windows to cluster into larger regions; finally, a '*stepwise optimal merging algorithm*' causes the further merger of clusters until a stopping condition is reached.

The initial splitting process takes a square window, denoted s and splits it into four if the distance between the model parameter estimates calculated across its entirety (i.e. the mean μ_s and the covariance matrix $\mathbf{\Sigma}_s$) and any of the model parameters estimated from each of the four segments, s_1, s_2, s_3 and s_4, exceeds a specific preset threshold (ϵ_μ or ϵ_σ). In the case of the two covariance matrices, the measure used is the L_2 norm.

Before describing the merging processes a measure must first be defined which enables the comparison of the homogeneity of data within windows before and after merging. The criteria chosen in [78] is a ratio of pseudo-likelihoods: specifically, if the merger of regions $k, l \rightarrow m$ is considered and $\mathrm{PL}(s_i \mid \mu_i, \mathbf{\Sigma}_i)$ denotes the pseudo-likelihood of the i'th region, then the required ratio is given by

$$\mathrm{R}_{\mathrm{PL}}(k,l) \;=\; \frac{\mathrm{PL}(s_k \mid \mu_k, \mathbf{\Sigma}_k)\;\mathrm{PL}(s_l \mid \mu_l, \mathbf{\Sigma}_l)}{\mathrm{PL}(s_m \mid \mu_m, \mathbf{\Sigma}_m)} \tag{3.11}$$

The resulting criteria for the merger are, firstly the combined region must be deemed homogeneous using a test similar to that used in the splitting step, and secondly the pseudo-likelihood ratio must be less than a pre-defined threshold. All regions in the image are iteratively considered for merger until no further merges are possible.

The final step, known as *stepwise optimal merging* continues the merging process. Regions are combined if they satisfy the following condition: $(k,l) = \arg\min_{k,l} \mathrm{R}_{\mathrm{PL}}(k,l)$. Thus, even though this can cause the joint pseudo-likelihood over the complete image to decrease, further merges are allowed. The process continues until a stopping criteria is met: i.e. until $(\mathrm{R}^{n+1}_{\mathrm{PL}} - \mathrm{R}^{n}_{\mathrm{PL}}) \geq \alpha$, where $\mathrm{R}^{n}_{\mathrm{PL}}(\cdot)$ is the pseudo-likelihood ratio for the n'th merge and α is a pre-defined constant.

The arbitrary setting of the various thresholds and a stopping parameter required by the segmentation process causes a large degree of supervision to be implicit in the algorithm's

implementation. However, the utilisation of spatial adjacency in the clustering process represents an advance over the previously discussed windowing-clustering algorithms.

3.3 Local Spatial Statistics

Texture features derived from local spatial gray-level statistics are among the most widely used throughout the texture segmentation literature, for example: crop discrimination using synthetic aperture radar imaging [87]; texture discrimination in ultrasound images [74], and more generally in application of remote sensing [48].

A representation of such statistics, coocurrence matrices, was brought to prevalence by Haralick *et al*[46] and Haralick[45]. Cooccurrence matrices consist of elements indicative of the relative frequency of occurrence of two gray-level values at a specific separation on the image lattice. To obtain spatial data at each pixel site within an image, cooccurrence matrices are calculated over a windowed region surrounding each pixel site. The need to generate several different matrices (dependent on the number of displacements being considered) at each pixel location within the image represents a massive processing task. However, as demonstrated by Hickman *et al*[48], by applying parallel processing techniques, the procedure may be speeded up considerably.

The generation of each matrix facilitates the calculation on many statistics. Several studies have been undertaken to find the most amenable subsets for particular types of problem. These include a study by Soares *et al*[87] where the applicability of such statistics was considered with reference to crop discrimination from SAR (synthetic aperture radar) images. The features considered in the study comprise energy, entropy, contrast, local homogeneity, correlation and Chi-square. If each element of the cooccurrence matrix is denoted $p(i,j)_\tau$, where τ indicates the displacement between the two pixels under consideration and i and j are the two gray-scale values in question, then the listed features

may be expressed as follows,

$$\text{Energy} = \sum_i \sum_j p(i,j)_\tau^2 \tag{3.12}$$

$$\text{Entropy} = -\sum_i \sum_j p(i,j)_\tau \log p(i,j)_\tau \tag{3.13}$$

$$\text{Contrast} = \sum_i \sum_j (i-j)^2 p(i,j)_\tau \tag{3.14}$$

$$\text{Local Homogeneity} = \sum_i \sum_j \frac{p(i,j)_\tau}{1+(i-j)^2} \tag{3.15}$$

$$\text{Correlation} = \sum_i \sum_j \frac{(i-\mu_1)(j-\mu_2)p(i,j)_\tau}{\sigma_1 \sigma_2} \tag{3.16}$$

$$\text{Chi-square} = \sum_i \sum_j \frac{p(i,j)_\tau^2}{p(i,\cdot)_\tau \; p(\cdot,j)_\tau} \tag{3.17}$$

where,

$$\begin{array}{llll} \mu_1 = \sum_i i \; p(i,\cdot)_\tau & & \mu_2 = \sum_j j \; p(\cdot,j)_\tau \\ \sigma_1 = \sqrt{\sum_i (i-\mu_1)^2 \; p(i,\cdot)_\tau} & & \sigma_2 = \sqrt{\sum_j (j-\mu_1)^2 \; p(\cdot,j)_\tau} \\ p(i,\cdot)_\tau = \sum_j p(i,j)_\tau & & p(\cdot,j)_\tau = \sum_i p(i,j)_\tau \end{array} \tag{3.18}$$

Although these measures are not mutually independent, (they generally can be considered correlated with each other) they do represent measures of meaningful and differing textural entities. Energy (or angular second moment) measures uniformity of texture. Entropy measures image disorder and is thus inversely correlated with energy. Contrast may be thought of as a measure of mean gray-level difference between pixels separated by the displacement τ. Contrast is also strongly uncorrelated with second order statistics, such as energy. Local homogeneity (or the inverse difference moment) measures uniformity of texture and thus may often be inversely correlated with contrast. Correlation measures any linear dependency between displaced pixel pairs and is thus typically uncorrelated with energy and entropy. Finally, the Chi-square statistic may be thought of as a pairwise energy term that has been normalised to relative occurrence of individual pixel values. It is therefore self-evidently correlated with energy.

The computational demands of calculating such features gives a practical concern: a trade off is required between the number of gray-scale levels considered (i.e. the dimension of the cooccurrence matrices) and the size of region from which each matrix is generated.

To allow a reasonably accurate measure of each region's statistics, the number of gray-scale levels must be significantly less than the number of pixels.

The family of local spatial statistics was extended by Geman *et al* [37] to incorporate further spatial statistics, what were termed isotropic and directional residuals. Such statistics were chosen on the basis of empirical evidence and thus are somewhat *ad hoc*. Examples of such statistics include: the isotropic residual obtained when comparing a central pixel value with that of the perimeter pixels of a window, and the directional residuals obtained when differencing a central pixel with its horizontally or vertically adjacent neighbours.

Local spatial statistics are often used but neglecting one fundamental concept: there is an effective trade-off between localisation of boundaries between textures and the size of window needed to generate reliable statistics to differentiate between textures. This problem lies at the heart of many image processing problems but with the exception of Gabor Filtering in the spatial-frequency domain (see section 3.4), has yet to identify formal or optimal solutions.

3.3.1 Image models based on Local Spatial Statistics

The incorporation of local spatial statistics into an image model has been achieved by several authors, most notably by Geman *et al* [37] and Kervrann & Heitz[61]. The approach generally adopted uses a derived cost function from the local spatial statistics, which is combined with the energy function of a Potts' model on the hidden states or segmentation. The maximum *a posteriori* segmentation of the associated Gibbs distribution may then found, thus casting the segmentation as an optimisation problem.

To elaborate further, a regular lattice $\mathbf{\Omega}^{(X)}$ is first specified, on which the underlying or hidden image labels are defined $\mathbf{x} = \{x_s, s \in \mathbf{\Omega}^{(X)}\}$. Each pixel is assigned to one of K states, hence $x_s \in \Lambda = \{0, 1, ..., K\}$. The observed image gray-scale values are defined on a different lattice $\mathbf{\Omega}^{(Y)}$; $\mathbf{y} = \{y_s, s \in \mathbf{\Omega}^{(Y)}\}$. The distinction between the two lattices is important: they will differ in resolution, with $\mathbf{\Omega}^{(X)}$ being specified at a coarser resolution than $\mathbf{\Omega}^{(Y)}$, thus allowing the assignment of segmentation labels to blocks of observed pixels, rather than to single pixels.

To compare neighbouring blocks of observed pixels, for example blocks $\mathbf{y}_{\mathbf{D}_s}$ and $\mathbf{y}_{\mathbf{D}_t}$ (where $\mathbf{D}_s$ denotes sites forming the block corresponding to site s in the coarser lattice $\mathbf{\Omega}^{(X)}$), Geman *et al*[37] introduce the disparity measure $\Phi_{s,t}(\mathbf{y}) \in \{+1, -1\}$. This utilises

the local spatial statistics, discussed in the previous section, to assess the degree of similarity between the two blocks of observed pixel data.

The disparity measure commonly used to define distance between two gray-scale distributions is the Kolmogorov-Smirnov statistic. To specify this, consider two sets of data $\mathbf{v}^{(1)} = \{v_1^{(1)}, v_2^{(1)}, ..., v_{n_1}^{(1)}\}$ and $\mathbf{v}^{(2)} = \{v_1^{(2)}, v_2^{(2)}, ..., v_{n_2}^{(2)}\}$, then two sample histograms may be formed $F_1(t)$ and $F_2(t)$, such that $F_i(t) = \frac{1}{n_i}\#\{k : v_k \leq t\}$ where $\#\{\cdot\}$ is the 'size of set' operator. The Kolmogorov-Smirnov distance can now be defined as simply the maximum distance between these two distributions over t;

$$d(\mathbf{v}^{(1)}, \mathbf{v}^{(2)}) \;=\; \underset{-\infty \leq t \leq +\infty}{max} \; |F_1(t) - F_2(t)| \tag{3.19}$$

The Kolmogorov-Smirnov statistic has an obvious invariance property: it is invariant to strictly monotonic transformations of the two sets of data, and is thus the independent of the underlying distributions. Since histograms of gray levels are often insufficient to distinguish between different textures, several transformations are employed (for example the residuals described in the previous section might be calculated) to extrapolate various spatial statistical properties from the pixel data. Denoting these transformations of the data $\mathbf{y} \Rightarrow \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(n)}\}$, the disparity measure $\Phi_{s,t}(\mathbf{y})$ can now be derived by forming a function of the individual Kolmogorov-Smirnov distance measures applied to these transformed data sets. This final disparity measure takes the value $+1$ if any of the n distances exceeds a preset threshold, c and -1 otherwise. More formally,

$$\Phi_{s,t}(\mathbf{y}) \;=\; 2\;\delta\Big([d(\mathbf{y}_{D_s}^{(1)}, \mathbf{y}_{D_t}^{(1)}) > c_1] \oplus [d(\mathbf{y}_{D_s}^{(2)}, \mathbf{y}_{D_t}^{(2)}) > c_2] \oplus .... \oplus [d(\mathbf{y}_{D_s}^{(n)}, \mathbf{y}_{D_t}^{(n)}) > c_n]\Big) \;-\; 1 \tag{3.20}$$

where $\oplus$ denotes the 'logical or' function and $\delta(\cdot)$ is the Kronecker delta function.

Kevrann & Heitz[61] adopt a different approach: they compare the distribution of a block of observed pixels with each of the distributions of pixel data as assigned by the current segmentation estimate. Hence, by using the current estimate underlying segmentation $\mathbf{x}$, histograms indicative of the pixel distributions corresponding to each of the different states may be generated. The comparison of histograms uses an identical methodology to that just described, hence Kevrann & Heitz's disparity measure may be defined $\Phi_s(\mathbf{y}, \mathbf{x}) \in \{+1, -1\}$.

To model interactions between segmented regions, thus promoting structure in the segmentation estimate, priors are assigned on the segmentation. These will typically take the form of a Potts model type MRF and thus require the definition of a neighbourhood structure $\boldsymbol{\eta}$, and the assignment of a corresponding potential function $\Psi_{s,t}(\mathbf{x})$, thus defining the

degree of coupling between adjacent segmentation variables. The sum of these potentials together with the defined disparity cost measure gives the Gibbs energy function associated with the posterior distribution for the Kevrann & Heitz model,

$$U_{KH}(\mathbf{x},\mathbf{y}) \;=\; \sum_{s\in S, t\in \boldsymbol{\eta}_s} \Psi^{KH}_{s,t}(\mathbf{x}) \;+\; \Phi_s(\mathbf{x},\mathbf{y}) \tag{3.21}$$

Geman *et al* use a similar regularisation function but define their posterior energy to be the sum accross the lattice of the products of the site specific potentials and disparity measure. This difference in energy function structure requires the potentials to take a different range: for the Geman *et al* model the function takes strictly positive, 'penalty', values but the Kervrann and Heitz model allows negative values to encourage particular clique formulations.

One final and novel concept, introduced in [37], is the application of a further penalty term $V(\mathbf{x})$ to the prior on $\mathbf{x}$. This additional term is a function of the number of occurrences in the image model realisation of forbidden configurations of states, usually defined to be both small regions and narrow regions. If λ governs the strength of such a term then the full their energy function is given by

$$U_G(\mathbf{x},\mathbf{y}) \;=\; \sum_{s\in S, t\in \boldsymbol{\eta}_s} \Psi^{G}_{s,t}(\mathbf{x})\; \Phi_{s,t}(\mathbf{y}) \;+\; \lambda V_s(\mathbf{x}) \tag{3.22}$$

3.3.2 Feature Based Relaxation Algorithms

Having described how local spatial statistics may be incorporated into an image model using smoothing functions of the types described in the previous chapter, the mechanisms by which they might be optimised are now examined.

Although not addressing the problem of estimating the optimal number of regions, Geman *et al* [37] developed an algorithm that allowed boundaries to be detected between regions of differing gray-scale or texture. This approach was extended further by Kervrann & Heitz[61] who developed a fully unsupervised segmentation algorithm.

As described in section 2.3 Geman & Geman[39] had already shown that Gibbs sampling and simulated annealing form a powerful tool for finding the global mode of a multivariate distribution, proving convergence to the global maximum as the temperature parameter of the distribution is slowly lowered to zero, but in infinite time.

However, the introduction of the penalty term $V(\mathbf{x})$ in equation 3.22, constrains the optimisation to a subset of all configurations, i.e. $\mathbf{X}^* \subset \mathbf{X}$ or more explicitly $\mathbf{X}^* = \{\mathbf{x}$:

$V(\mathbf{x}) = \min_{\mathbf{x}'} V(\mathbf{x}')\}$. This reduces the optimisation to the maximisation of

$$\Pi^*(\mathbf{x}) = \delta(\mathbf{x} \in \mathbf{X}^*) \frac{\exp\{-U(\mathbf{x})\}}{\sum_{\mathbf{x}' \in \mathbf{X}^*} \exp\{-U(\mathbf{x}')\}} \tag{3.23}$$

By constructing a non-homogeneous Markov chain $\{\mathbf{x}_k, k = 0, 1, ...\}$ through the use of the Gibbs sampler and the target distribution

$$\Pi_k(\mathbf{x}) = \frac{\exp\{-\frac{1}{T_k}[U(\mathbf{x}) + \lambda_k V(\mathbf{x})]\}}{\sum_{\mathbf{x}' \in \mathbf{X}^*} \exp\{-\frac{1}{T_k}[U(\mathbf{x}') + \lambda_k V(\mathbf{x})]\}} \tag{3.24}$$

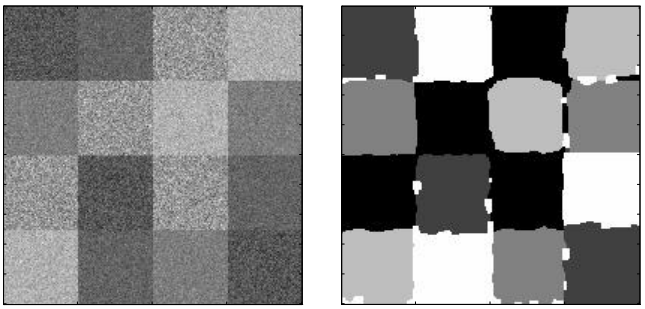
and by defining schedules for temperature and penalty function parameters so that $t_k \searrow 0, \lambda_k \nearrow \infty : \frac{\lambda_k}{t_k} \leq \alpha \log k$, where α is an arbitrary constant and k indicates the iteration number, Geman *et al* [37] showed that the asymptotic distribution of $x(k)$ will be uniform over the solution set of the constrained optimisation problem. Alternatively, by defining the two schedules as prescribed, then as $k \to \infty$, the sampler will draw from a set of values comprising just those which satisfy the constrained optimisation problem.

This constraining of the energy minimisation of optimisation is extremely useful since it removes many local minima from the energy surface. These minima are typically caused by the symmetry of the neighbourhood structure $\boldsymbol{\eta}$, as encountered previously in section 2.5.5. For example, if one prescribes a nearest neighbourhood structure to the prior model then self-evidently, if two of the neighbours of a pixel belong to one state and the other two to another, then our prior knowledge says the central pixel is equally as likely to belong to one state as the other. Such a conditioning on a nearest neighbourhood causes configurations comprising 4×4 pixel blocks to be extremely stable. This is an inherent problem for deterministic optimisation algorithms but is not necessarily fatal in stochastically driven algorithms, e.g. simulated annealing. However, the incorporation of Geman *et al*'s [37] penalty term can remove these anomalies, thus improving convergence of an optimisation algorithm.

To achieve unsupervised segmentation Kervrann & Heitz[61] augment the state space $\boldsymbol{\Lambda}$ with an additional state ρ, which identifies an class specifically for outlier data. Thus if the state-space currently comprises K states, then the augmented state-space will be $\Lambda = \{0, 1, ...K-1, \rho\}$. Conditional energy functions for each of the first K states may be formed from equations 3.21 or 3.22 and will be of similar form to

$$U(x_s \mid \mathbf{x}_{\eta_s}, \mathbf{y}_{\mathbf{D}_s}) = \Phi_s(x_s, \mathbf{y}_{\mathbf{D}_s}) + \beta\, \Psi^{KH}_{s,t}(x_s \mid \mathbf{x}_{\eta_s}) + \lambda\, V(x_s \mid \mathbf{x}_{\eta_s}) \tag{3.25}$$

However, for the outlier class a probability measure indicative of a large disparity between the histogram of surrounding states and each of the current state histograms is required.



(a) (b)

Figure 3.4: *(a) Gray scale pattern with colour dependent additive Gaussian noise, (b) segmentation.*

This can be defined by a parameter ϕ, which governs the likelihood of the allocation of pixels to the outlier class. The resulting conditional energy is given by

$$U(x_s \mid \mathbf{x}_{\eta_s} \mathbf{y}_{\mathbf{D}_s}) \;=\; \beta\, \Psi_{s,t}^{KH}(x_s \mid \mathbf{x}_{\eta_s}) \;+\; \lambda\, V(x_s \mid \mathbf{x}_{\eta_s}) \;+\; \phi \tag{3.26}$$

Thus a large value of ϕ indicates a high level of disparity needs to be incurred between the histogram of surrounding pixel data and that of all the state histograms before a pixel is allocated to the outlier class. Alternatively, a high value of ϕ disfavours the creation of new states.

The incorporation of an outlier class facilitates the implementation of a somewhat *ad hoc* unsupervised optimisation process. Explicitly: begin with a single state, i.e. set $K = 1$; assign a high value to the temperature parameter; set the penalty parameter λ to a low value; then, carry out an annealing process to segment the image into states 0 and ρ. If any of the connected regions labelled by ρ cover more than a pre-designated area of the image then assign them to a new state, i.e. 1, (thus increasing $K \rightarrow 2$) but leave any remaining small regions assigned to the outlier class ρ.

This process can be repeated iteratively, resetting the temperature and penalty parameters at each stage before carrying out fresh annealing processes. When no new states are generated, the algorithm can be expected to have converged and any small outlier regions can be removed by carrying out a further annealing run after randomly reallocating these sites to the existing K classes.

Such an algorithm is difficult to implement. Many parameters have to be pre-specified, for example: all parameters implied in the specification of the annealing schedules; the

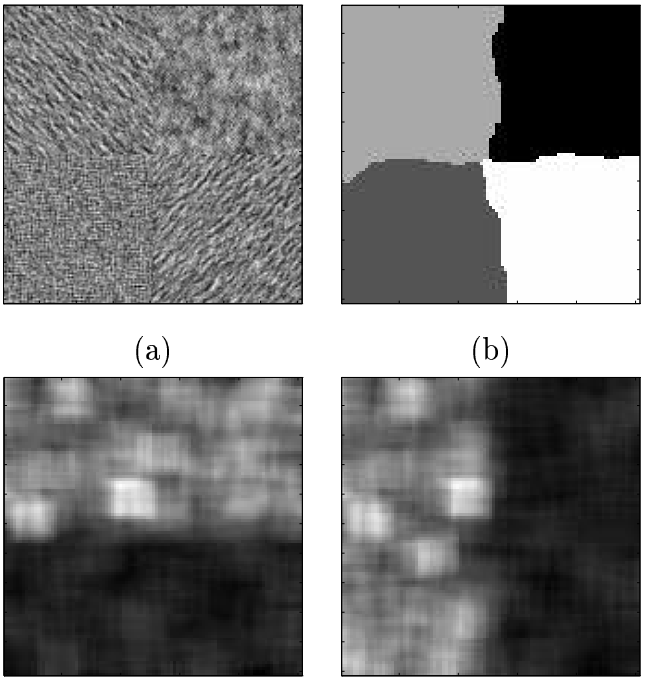


Figure 3.5: *(a) texture mosaic synthesised using a GMRF model, (b) associated segmentation based on contrast transforms, (c) contrast transform based on the cooccurence matrix defined over translation* $[+1, +1]$*, (d) contrast transform based on the cooccurence matrix defined over translation* $[+1, -1]$

size of regions over which cooccurrence matrices and other features are generated; the size of regions over which the KS distances are calculated; the values of the disparity measure thresholding parameters $\{c1, c2, ..., c_n\}$ (from equation 3.20); the parameter ϕ, governing the probability of assignment to the outlier class, and finally, the exact specification of penalty function. Many of these parameters are hard to specify due to the final optimisation being extremely sensitive to their value.

The performance of these algorithms is demonstrated in figures 3.4 and 3.5. Parameter values used in the first experiment (figure 3.4) were $\beta = 1.2$, $\phi = 0.2$ and with the threshold being set to 0.12 for the KS distance on the raw gray-scale data. For the second experiment, (figure 3.5), two transforms of the gray-scale image, both of contrast (given previously by equation 3.15) but from different cooccurrence matrices generated over the two transforms $[+1, +1]$ and $[+1, -1]$ were used. The associated parameter values were $\beta = 1.3$, $\phi = 0.8$, with thresholds c_1 and c_2 being both set to 0.45 against the KS distances generated from

each of the two sets of transformed data. In both experiments the penalty term was defined over 7×7 pixel blocks and a penalty was incurred if less than one quarter of these pixels matched the central state.

These results demonstrate the great weakness of all segmentation algorithms based on derived features: the necessity for windowing causes discrepancies to occur at region boundaries because of an implied trade-off between the accuracy of texture identification and the precision of their spatial localisation. An approach designed to be jointly optimal in terms of these conflicting criteria is described in the following section.

3.4 The multi-channel filtering approach to texture segmentation

The multi-channel filtering approach to texture segmentation has been developed from our understanding of the human early vision system. This has evolved through a series of psychophysical experiments. Initial models of the early vision system were postulated on the basis on the assumption that the process was composed simply of feature detection in the spatial domain.

Campbell and Robson[17] were the first to propose a multi-channel model of the early vision system. They found that the level at which the human visual system could differentiate between square and sinusoidal gratings was far lower than would be expected if the vision system processed all frequencies similarly. They thus realised that the level at which differentiation occurred corresponded to that at which the higher harmonics reach their own threshold and hence they envisaged a system at which channels acted effectively like bandpass filters.

De Valois *et al*[95] presented data showing that macaque monkey cortical cells have localised responses in the two-dimensional spatial-frequency domain, making them orientation selective. Daugman[27] used this to show that existing theories for receptive field profiles did not match this data and then suggested that a sinusoidal signal modulated by a Gaussian envelope might possess the desired properties.

Marcelja[71] showed that a one-dimensional Gabor function fitted the response of a

single cell of the visual cortex in both spatial and spatial-frequency domains. Gabor's[33] fundamental result was quoted, showing that the product of a one dimensional signal's time duration and bandwidth, known as its joint uncertainty, is limited by a lower bound (see Papoulis[79] for a proof) and that this lower bound is only reached by a particular family of signals which comprise a sinusoidal signal modulated by a Gaussian envelope. The importance of this feature to the visual cortex was first recognised by Marcelja: 'the simultaneous maximal localisation in space and spatial-frequency'. This resolved the ill-founded debate between parties arguing that the visual system comprised a linear process in either the spatial or spatial-frequency domain, since the Gabor transform would cause a linear process in one domain to imply a linear process in the other.

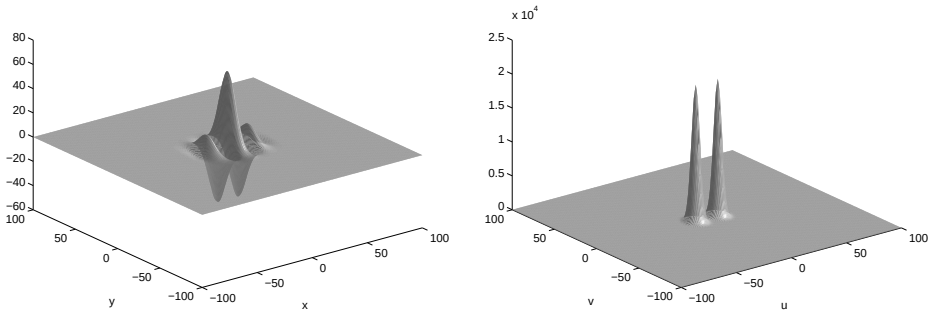
Daugman[28] extended Gabor's results to two dimensions establishing a two-dimensional uncertainty principle and finding the family of functions that reach the associated lower bound, explicitly a sine wave modulated by a two dimensional Gaussian envelope.

These results are crucial to the problem of texture segmentation. To achieve reliable texture segmentations through filtering, functions that are short in spatial duration are required to allow accurate localisation of boundaries. However, to differentiate between similar textures, it is desirable to use filters with small bandwidths. By using the two-dimensional Gabor functions derived by Daugman[28], the optimum trade-off between variance in the spatial and spatial-frequency domains is reached.

Turner[94] used Gabor filter banks to discriminate between areas of different texture. Filters were chosen with different orientations and frequencies to select features in the spatial-frequency domain of the differing textures. Turner [80] also obtained an important result showing that neighbouring cortical cells comprised identical Gabor functions but differing by 90° in phase. It is apparent that by forming the Euclidean sum of the two filter outputs it is possible to generate a phase insensitive filtered image.

A thorough review of texture segmentation using Gabor filter banks was undertaken by Bovik *et al.*[15]. Various patterns of filter banks on the spatial-frequency domain were tried and the filtered images were improved by post-filtering the outputs using a two-dimensional Gaussian filter to remove ripples induced by borders between textures in the original image.

Several formulations of Gabor filters have been used to perform texture segmentation. Bovik *et al.*[15] used complex Gabor filters. The two-dimensional impulse response of an even-symmetric Gabor filter defined on a plane whose co-ordinate system is denoted (x, y),



(a) (b)

Figure 3.6: *(a) Impulse response and (b) spatial-frequency response of a Gabor filter with 0 ° orientation and centred at 7 cycles per image width.*

is given by

$$h(x,y) \;=\; \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right]+2\pi i u_0 x\right\} \tag{3.27}$$

which comprises the modulation of a complex planar sinusoidal waveform, frequency u_0, by a two-dimensional Gaussian envelope defined by the parameters σ_x and σ_y.

The two-dimensional Fourier transform yields the spatial-frequency domain representation which is often referred to as the modulation transfer function since it defines the scaling or modulation of each frequency component occurring in the input image. It is given by

$$\begin{aligned} H(u,v) \;=\; \frac{1}{2\pi\sigma_u\sigma_v}\Bigg(&\exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\} \;+ \\ &\exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\}\Bigg) \end{aligned} \tag{3.28}$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$.

Jain & Farrokhnia[52] use symmetric real valued filters comprising the real or cosine part of a complex Gabor spatial impulse response function. This reduces the number of filters required by a factor of two, however the filter outputs are no longer phase insensitive so post-filtering with a Gaussian filter becomes essential. The spatial and spatial-frequency domain impulse responses of such a real valued Gabor filter are shown in figure 3.6.

Multi-channel filtering of an input image requires a filter bank capable of extracting various orientations and frequency components occurring within the image. Several authors have approached this problem by covering the spatial-frequency domain with various

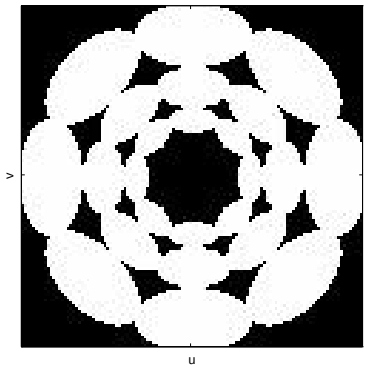


Figure 3.7: *Half peak spatial-frequency domain coverage of a set of 12 daisy petal Gabor filters.*

daisy petal patterns[52][81][69][29], where each 'petal' consists of a Gabor filter modulation transfer function (see figure 3.7).

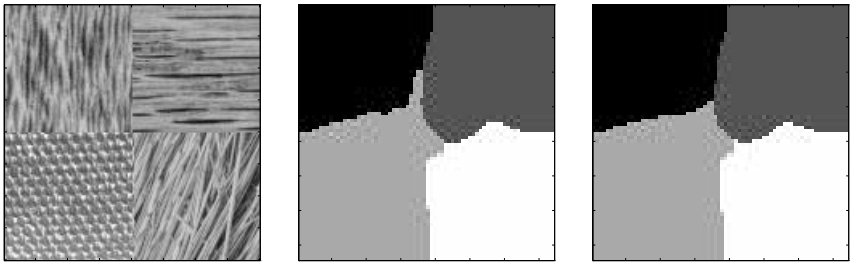
Such an arrangement is arrived at by considering each filter's half-peak frequency bandwidth B_r, and orientation bandwidth B_θ, each given by

$$B_r = \log_2\left(\frac{u_0 + (2\ln 2)^{\frac{1}{2}}\sigma_u}{u_0 - (2\ln 2)^{\frac{1}{2}}\sigma_u}\right) \tag{3.29}$$

$$B_\theta = 2\tan^{-1}\left(\frac{(2\ln 2)^{\frac{1}{2}}\sigma_v}{u_0}\right) \tag{3.30}$$

Here the frequency bandwidth is defined in octaves. The allocation of centre frequencies for neighbouring rings of filters has typically been specified in an *ad hoc* manner. For example, Jain & Farrokhnia [52] set the ratio between centre frequencies of each ring of filters to a value of two. The number of different orientations chosen varies between four [15][52] to six [69].

Post-filter processing is required to remove ripples induced by the convolution of the filter's impulse response function with edges and boundaries within the input image. Jain & Farrokhnia [52] used a nonlinear function which effectively saturates or thresholds the filter output, minimising any ripple. The resulting image is then divided into overlapping windows and for each window, the absolute mean or 'texture energy' is computed. The window size is critical since the larger the window the more exact the energy measure, but the less accurate the texture boundary localisation. A Gaussian window is suggested whose spatial constant is given by, $\sigma = 0.5N/u_0$, where N is the image width.



(a) (b) (c)

Figure 3.8: *(a) Brodatz Texture Mosaic, (b) result of K-means segmentation without including (x, y) coordinates in the feature vector, (c) result with the inclusion of (x, y) coordinates.*

To complete the segmentation algorithm, the filtered images can be used to form elements of feature vectors. The vectors may then be used as inputs to a classification algorithm, for example, Jain & Farrokhnia [52] use a variant of the K-means algorithm. A major problem with such a technique occurs when boundaries within the image and slight non-homogeneities in the textures cause spurious miss-classifications. To overcome this, Jain & Farrokhnia [52] extend the feature vector to include (x, y) coordinate information, thus ensuring their algorithm favours circular regions. This improves results for typical Brodatz texture mosaic tests (as can be seen from figure 3.8) but it is easy to see that such a bias would make the processing of real-world images impracticable.

3.5 Conclusions

The segmentation techniques discussed throughout this chapter have relied upon the use of various features indicative of gray-scale statistical and textural local content. The range of features is diverse: model based statistics; local spatial statistics, for example those derived from coocurrence matrices, and also statistics derived in the spatial-frequency domain.

Algorithms that utilise these features take two general forms. Either clustering techniques are applied, which in the unsupervised case requires the pre-specification of at least one thresholding parameter, or a regularising Potts model is optimised whose energy function includes what is effectively an external field component, dependent on such features.

The algorithms appear insufficient for satisfying the most general of unsupervised criteria. There are several reasons for this: firstly the use of windows to calculate features will mean that small outlier regions will be unrepresented in feature space; the implied trade-off between window size and localisation of region boundaries is generally sub-optimal; finally, many clustering algorithms fail to utilise fully the spatial relationship between their associated windows in the image.

However, these algorithms often prove far less computationally intensive than their model-fitting counterparts of the previous chapter. The generation of features is often an efficient process, thus an important theme of chapter 5 is the specification of a mechanism by which such features might be used to speed up both a model selection and optimisation process.

Unsupervised Segmentation Algorithms 4

4.1 Introduction

The approach to unsupervised segmentation presented here[1] comprises a Markov Chain Monte Carlo (MCMC) algorithm to sample from the posterior distribution so that simulated annealing may be used to estimate the MAP solution. This methodology is similar to that used in [39] to segment an image using an hierarchical MRF model with parameters known *a priori*. Here the method is extended so that the sampling scheme and hence the MAP estimate is not just over the segmentation of the image into classes, but also the number of classes and their respective model parameters.

For simulated annealing convergence criteria to be met, a sampling algorithm is required that is capable of exploring the entire model space. More formally the associated Markov Chain should be irreducible with respect to the target distribution, see section 2.3 for more details. Hence, to achieve optimisation over a set of models the sampler will need to move throughout the combined model space, i.e. jumping between models. The reversible jump algorithm [43] is used for this purpose and it is introduced here in the context of the more general model selection problem.

The unsupervised segmentation algorithms presented in this chapter have advantages over those feature based algorithms reviewed in chapter 3: no windowing is required to estimate region model parameters, thus there is no limitation on the minimum size and shape of pictorial elements; there is also no trade-off between accuracy of spatial localisation

[1] The work included in this chapter was presented at the 'International Workshop EMMCVPR'97, Venice, Italy' and sections concerning the unsupervised segmentation of Isotropic MRF's were published in, 'Lecture Notes in Computer Science' [5]. Additional sections concerning unsupervised texture segmentation using Gaussian MRF models will appear in, 'Pattern Recognition' [4].

of features and discrimination between textures; no thresholding is required thus there is no necessity to set *ad hoc* prior parameter values; finally, spatial data is inherent in the Bayesian model framework and as the algorithm presented here forms a direct optimisation process for this model, this information is utilised fully, which was not the case in the various feature based clustering algorithms considered previously.

The algorithm's MCMC methodology removes the necessity for an exhaustive search over a subsection of the parameter space. This ensures an improvement in efficiency over algorithms described in section 2.6: these require separate optimisations to be carried out for each model before a model order selection is made. The process used here also eliminates the necessity for the use of information criteria for the comparison of models. Here the difference in model spaces and measures is compensated for in the sampling algorithm and thus direct comparison of posterior probabilities is possible. Also, the structure of the posterior distribution allows the natural integration of any prior knowledge concerning model order.

The remainder of this chapter is divided as follows: the next section introduces the problem of Bayesian model selection applied to the hidden data problem; section 4.3 describes how the reversible jump sampling algorithm can be utilised to explore the corresponding posterior distribution; section 4.4 defines the image models used throughout the chapter; the posterior distributions for the noisy image and texture models are derived in sections 4.4.1 and 4.4.2; section 4.5 describes the algorithms employed to sample from these distributions, the segmentation process or allocation of class labels to pixel sites is given, as is the sampling scheme for noise and MRF model parameters from their conditional densities; the method by which reversible jumps are incorporated into Markov Chains to enable sampling of the number of classes into which an image might be segmented is then described and this process is detailed for both noisy and textured image modes; experimental results for the resulting algorithms are presented in section 4.6 and the chapter is concluded in section 4.7.

4.2 Bayesian Model Selection and the Hidden Data Problem

The problem of unsupervised segmentation using a hierarchical MRF model is a subset of the group known as hidden data problems [92]. In section 2.6 and throughout the previous chapter non-Bayesian unsupervised segmentation strategies have been examined. These have chiefly consisted of using information criteria to compare different models or using various clustering algorithms. All of these existing methodologies require *ad hoc* prior information, (in the case of information criteria the choice of their exact form has proved somewhat subjective while the unsupervised clustering algorithms require various thresholds to be specified *a priori*) thus it would appear beneficial to pose the unsupervised segmentation problem in a Bayesian framework, where any prior information can be naturally combined in the model's general form.

By way of introduction, before examining how model selection can be posed in a Bayesian framework and how model selection might be implemented in the case of the hidden data problem, the somewhat simpler case of parameter estimation is first outlined.

Bayesian parameter estimation concerns the problem of finding the *best fit* parameter values to some observed data, given a specific model. To achieve this, an initial assumption concerning the choice of a criterion for such a selection must be made. If the primary objective is to choose a parameter estimation process that minimises the average probability of miss-classification then by application of the zero-one utility function[2] the Bayes decision procedure will be to choose the parameters that maximise the *a posteriori* probability, i.e. the maximum *a posteriori* (MAP) estimate.

More specifically, consider the case where: the model structure, $\mathcal{M}$ is known and parameterised by $\boldsymbol{\psi}$; data is observed and given by $\mathbf{y}$; a likelihood $p(\mathbf{y} \mid \boldsymbol{\psi}, \mathcal{M})$ and a prior density $p(\boldsymbol{\psi} \mid \mathcal{M})$ are specified; then the problem of MAP parameter estimation is given by

$$\hat{\boldsymbol{\psi}} \;=\; \arg\max_{\boldsymbol{\psi}\in\boldsymbol{\Psi}} \pi(\boldsymbol{\psi} \mid \mathcal{M}, \mathbf{y}) \tag{4.1}$$

where

$$\pi(\boldsymbol{\psi} \mid \mathcal{M}, \mathbf{y}) \;=\; \frac{p(\mathbf{y} \mid \boldsymbol{\psi}, \mathcal{M})\; p(\boldsymbol{\psi} \mid \mathcal{M})}{\int_{\boldsymbol{\psi}} p(\mathbf{y} \mid \boldsymbol{\psi}, \mathcal{M})\; p(\boldsymbol{\psi} \mid \mathcal{M})\; d\boldsymbol{\psi}} \tag{4.2}$$

[2] i.e. the cost of choosing the correct parameters is zero, but choosing incorrect parameters costs unity

The denominator of the above equation forms the predictive distribution $p(\mathbf{y} \mid \mathcal{M})$ and can in general be ignored when considering the task of MAP parameter estimation. More importantly however, the predictive probability provides a basis for assessing the compatibility of the observed data with the assumed model $\mathcal{M}$ and thus with the modeller's predictive beliefs. With this observation the problem of model selection can now be addressed.

To begin, the optimisation problem must first be framed: thus adopt the notation $\{\mathcal{M}^{(k)}, k \in \boldsymbol{\kappa}\}$ to denote the finite set of possible models, $\boldsymbol{\kappa}$, then for every model define an associated parameter vector and space, $\boldsymbol{\psi}^{(k)} \in \boldsymbol{\Psi}^{(k)}$; the resulting possible MAP model selection criteria are

$$\hat{k}, \hat{\boldsymbol{\psi}}^{(\hat{k})} = \arg \max_{k \in \boldsymbol{\kappa}, \boldsymbol{\psi}^{(k)} \in \boldsymbol{\Psi}^{(k)}} \pi(\boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)} \mid \mathbf{y}) \tag{4.3}$$

$$\hat{k} = \arg \max_{k \in \boldsymbol{\kappa}} \int_{\boldsymbol{\psi}^{(k)} \in \boldsymbol{\Psi}^{(k)}} \pi(\boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)} \mid \mathbf{y}) d\boldsymbol{\psi}^{(k)} \tag{4.4}$$

To evaluate these expressions, the posterior density $p(\boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)} \mid \mathbf{y})$ is required;

$$\pi(\boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)})\ p(\boldsymbol{\psi}^{(k)} \mid \mathcal{M}^{(k)})\ p(\mathcal{M}^{(k)})}{\sum_{k \in \boldsymbol{\kappa}} \int_{\boldsymbol{\psi}^{(k)} \in \boldsymbol{\psi}^{(k)}} p(\mathbf{y} \mid \boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)})\ p(\boldsymbol{\psi}^{(k)} \mid \mathcal{M}^{(k)})\ p(\mathcal{M}^{(k)})\ d\boldsymbol{\psi}^{(k)}} \tag{4.5}$$

To apply this paradigm to the problem of unsupervised segmentation using a hierarchical model requires the definition of a hidden state matrix $\mathbf{x}$, on which the segmentation will be defined. Since the ultimate goal is to achieve an optimal segmentation then the joint MAP estimate criterion of equation 4.3 is preferred. The resulting optimisation criterion is therefore,

$$\hat{k}, \hat{\boldsymbol{\psi}}^{(\hat{k})}, \hat{\mathbf{x}} = \arg \max_{k \in \boldsymbol{\kappa}, \boldsymbol{\psi}^{(k)} \in \boldsymbol{\Psi}^{(k)}, \mathbf{x} \in \mathbf{X}} \pi(\mathbf{x}, \boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)} \mid \mathbf{y}) \tag{4.6}$$

where the posterior distribution is given by

$$\pi(\boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)}, \mathbf{x} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)})\ p(\mathbf{x} \mid \boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)})\ p(\boldsymbol{\psi}^{(k)} \mid \mathcal{M}^{(k)})\ p(\mathcal{M}^{(k)})}{\sum_{k \in \boldsymbol{\kappa}} \int_{\boldsymbol{\psi}^{(k)} \in \boldsymbol{\psi}^{(k)}} \sum_{\mathbf{x} \in \mathbf{X}} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)})\ p(\mathbf{x} \mid \boldsymbol{\psi}^{(k)}, \mathcal{M}^{(k)})\ p(\boldsymbol{\psi}^{(k)} \mid \mathcal{M}^{(k)})\ p(\mathcal{M}^{(k)}) d\boldsymbol{\psi}^{(k)}} \tag{4.7}$$

The examination or estimation of this posterior distribution with a view to achieving the optimisation described in equation 4.6 is the subject of the remainder of this chapter.

4.3 Reversible Jump MCMC and the Hidden Data Problem

To sample from the posterior distribution given in equation 4.7 a mechanism is required that allows movement over the combined model space. The sampling process is an extension of that of the well known Metropolis-Hastings algorithm.

The Metropolis-Hastings algorithm is dynamic Monte Carlo method whose transition kernel is constructed to be both reversible and to ensure the existence of a pre-specified invariant distribution. Since detailed balance is a sufficient to ensure reversibility and the existence of an invariant distribution, it forms the basis of the Metropolis-Hastings algorithm.

If $\pi(x)$ is the desired target distribution and $p(x, dx')$ is the probability density function for the transition from $x \rightarrow x'$, then detailed balance is the condition such that, $\int_A \pi(dx) \int_B p(x, dx') = \int_B \pi(dx') \int_A p(x', dx)$. As shown in Appendix B, if the transition kernel is specified, $p(x, dx') \triangleq q(x, dx')\alpha(x', x)$ where $q(x, dx')$ represents a proposal density for the transition from $x \rightarrow x'$ and $\alpha(x', x)$ is an acceptance probability for that move, then the condition for detailed balance can be satisfied. To achieve this the acceptance probability or *acceptance ratio* must be defined so that

$$\alpha(x', x) = \min\left[1, \frac{\pi(dx')q(x', dx)}{\pi(dx)q(x, dx')}\right] \tag{4.8}$$

It should be noted that to calculate the acceptance ratio the probability of proposing the reverse transition $q(x, dx')$ must be calculable. This limits possible proposal distribution functions and has important repercussions when considering the model selection.

To apply this algorithm to the problem of model selection and in particular, the hidden data problem, requires the ability to move between model spaces and more specifically: the need for comparison of models defined on model spaces of differing dimension; compensation for changes in probability measure associated with these different model spaces and finally, a method by which the hidden data may be re-allocated to incorporate a change in state space. The resulting algorithm is the reversible jump sampler developed by Green [43].

The reversible jump sampling algorithm will now be outlined, however a proof of de-

tailed balance for the algorithm is presented in appendix B. If we consider the unsupervised hidden data problem or unsupervised hierarchical image model optimisation problem, as defined by equation 4.6 then a sampler is required that is capable of exploring the posterior distribution, equation 4.7.

The reversible jump sampling algorithm is comprised of various birth and death processes which facilitate the movement between model spaces. If the current label space $\mathbf{\Lambda} = \{0, 1, 2, ..., n-1\}$ then a birth will increase this space to $\mathbf{\Lambda} = \{0, 1, 2, ..., n-1, n\}$ and a death will decrease it to $\mathbf{\Lambda} = \{0, 1, 2, ..., n-2\}$. With these changes the parameter space will also expand or shrink in dimension. The reversible jump sampler of Richardson and Green [82] which is applied to the problem of estimating the number of components to a Gaussian mixture distribution, given some observed data represents a particular hidden data problem. Here births are treated as a splitting processes (i.e. a particular Gaussian is split into two) and deaths as mergers (i.e. two Gaussian distributions are merged to form one). There is no inherent reason why moves should be limited to this particular subset, but justification for their selection might be through their comparative ease of implementation and through their effectiveness as observed in experimental evidence.

The splitting of one state c into two, $c1$ and $c2$ requires the expansion of the parameter vector $\boldsymbol{\psi} = [\boldsymbol{\psi}_0, \boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_c, ..., \boldsymbol{\psi}_{n-1}] \rightarrow \boldsymbol{\psi}^+ = [\boldsymbol{\psi}_0, \boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_{c1}, \boldsymbol{\psi}_{c2}, ..., \boldsymbol{\psi}_n]$. Since these parameter vectors are of differing dimension there is no possibility of directly constructing a Markov transition kernel to implement such a move. The solution is to pad these vectors with extra components which take the form of random variables. Hence if $\mathbf{e}$ and $\mathbf{e}^+$ represent two random vectors and $\#(\cdot)$ is the operation '*dimension of*', then the padded vector must be such that, $\#\big([\boldsymbol{\psi}, \mathbf{e}]\big) = \#\big([\boldsymbol{\psi}^+, \mathbf{e}^+]\big)$.

The next step is to construct an invertible function to map between extended parameter spaces; $[\boldsymbol{\psi}^+, \mathbf{e}^+] = f\big([\boldsymbol{\psi}, \mathbf{e}]\big)$ and $[\boldsymbol{\psi}, \mathbf{e}] = f^{-1}\big([\boldsymbol{\psi}^+, \mathbf{e}^+]\big)$. Thus the new parameters $\boldsymbol{\psi}^+$ may be generated by drawing the random variables $\mathbf{e}$ from their respective distributions and substituting them, together with the old parameter values $\boldsymbol{\psi}$ into $f(\cdot)$. The mapping function between old and new parameter spaces must be such that the Radon-Nykodym Derivative [13] of the two measures be non-zero, i.e. $\left|\frac{\partial(\boldsymbol{\psi}^+, \mathbf{e}^+)}{\partial(\boldsymbol{\psi}, \mathbf{e})}\right| > 0$.

When the new parameters have been generated, the hidden data, labelled with state c needs to be re-allocated using a random sampling approach to $c1$ or $c2$. The procedure is limited to the selection from these two states since in the calculation of the acceptance ratio for the sampler, the reverse transition probability must be calculable; in other words,

if we were considering a merge process from $c1, c2 \rightarrow c$ then we would have to be able to calculate the probability of the opposing split; this would only be possible if the splitting process was limited to assigning states $c1$ or $c2$ in a random manner.

The preceding steps fully describe the proposal generation step of the reversible-jump algorithm when applied to the hidden data problem. The acceptance ratio is now calculated as in the Metropolis-Hastings algorithm. However, there are some extra terms the Radon-Nikodym Derivative (taking the form of a Jacobian Determinant) is included which compensates for the difference in probability measures between the proposed and original models. If we proceed with the above notation, then if $\mathbf{y}$ is the observed data, $\mathbf{x}$ and $\mathbf{x}^+$ are the two hidden data vectors before and after the split, then the acceptance ratio for the split is given by

$$\min\left[1, \frac{\pi^+(d\mathbf{x}^+, d\boldsymbol{\psi}^+ \mid \mathbf{y})}{\pi(d\mathbf{x}, d\boldsymbol{\psi} \mid \mathbf{y})} \frac{q(\mathbf{x}^+, d\mathbf{x} \mid \boldsymbol{\psi})}{q(\mathbf{x}, d\mathbf{x}^+ \mid \boldsymbol{\psi}^+)} \frac{q(d\mathbf{e}^+)}{q(d\mathbf{e})} \frac{q(\text{merge})}{q(\text{split})} \left|\frac{\partial(\boldsymbol{\psi}^+, \mathbf{e}^+)}{\partial(\boldsymbol{\psi}, \mathbf{e})}\right|\right] \tag{4.9}$$

where: the first term comprises the ratio between posterior probabilities; the second, that between the reverse and forward hidden data allocation probabilities; the third, the ratio of the probability of generating the random vectors corresponding to the reverse and forward mapping functions; the fourth, the ratio of the probability of merging over that of splitting, and the final term is the Radon-Nikodym Derivative between the two model probability measures.

To summarise, each splitting iteration of the reversible jump algorithm will consist of six steps,

---

1. generate the two new parameter vectors $\boldsymbol{\psi}_{c1}, \boldsymbol{\psi}_{c2}$ by drawing $\mathbf{e}$ from the respective proposal distribution and applying the function $[\boldsymbol{\psi}^+, \mathbf{e}^+] = f([\boldsymbol{\psi}, \mathbf{e}])$,
2. calculate the ratio $\frac{q(d\mathbf{e}^+)}{q(d\mathbf{e})}$ from the values obtained in step(1),
3. calculate the Radon-Nikodym Derivative of $f(\cdot)$ using the values obtained step(1),
4. reallocate the hidden data labelled with state c to the new classes $c1$ and $c2$ according to the distribution $q(\mathbf{x}, d\mathbf{x}^+ \mid \boldsymbol{\psi}^+)$,
5. using the values obtained in the previous step, calculate the ratio $\frac{q(\mathbf{x}^+, d\mathbf{x}|\boldsymbol{\psi})}{q(\mathbf{x}, d\mathbf{x}^+|\boldsymbol{\psi}^+)}$,
6. calculate the overall acceptance ratio according to equation 4.9 and accept or reject the proposal based upon this value.

When considering a merge, the same process is followed but applying the opposite transitions or calculations at each step. The acceptance ratio is consequently the inverse of that of equation 4.9.

4.4 Image Models

This section introduces the notation for the Hierarchical Markov Random Field image models used throughout the remainder of this dissertation.

Let $\mathbf{\Omega}$ denote an $M \times N$ lattice indexed by (i, j) so that $\mathbf{\Omega} = \{(i, j); 1 \leq i \leq M,\ 1 \leq j \leq N\}$. Let $[\mathbf{Y} = \mathbf{y}] \equiv \{Y_s = y_s;\ s \in \mathbf{\Omega}\}$ be the observed gray-scale image where pixels take values from the interval $(0, 1]$. Then let $[\mathbf{X} = \mathbf{x}] \equiv \{X_s = x_s;\ s \in \mathbf{\Omega}\}$ correspond to the labels of the underlying Markov Random Field which have values taken from $\mathbf{\Lambda} = \{0, 1, ..., k-1\}$. This notation is lengthy and so will be shortened for convenience to $\mathbf{x} \equiv \{x_s;\ s \in \mathbf{\Omega}\}$, from now on.

If $\boldsymbol{\eta}_s$ defines a neighbourhood structure at site s, then let the vector of random variables which are the labels that comprise that neighbourhood be $\mathbf{x}_{\eta_s}$. Similarly, let $\boldsymbol{\rho}_s$ define a second, usually different neighbourhood structure at site s but let this be defined on the observed image $\mathbf{y}$, so that $\mathbf{y}_{\boldsymbol{\rho}_s}$ is the vector of pixel gray-scale values over that neighbourhood. Further, let all model parameters be included in the parameter vector $\boldsymbol{\psi}$. If a Gibbs distribution is used to model the likelihood of observing the image $\mathbf{y}$ given the label field $\mathbf{x}$, and to model all *a priori* knowledge of spatial correlations within the image and label fields then the conditional posterior distribution for an observed pixel gray-scale value and class label at site s, given the two neighbourhoods is,

$$p(y_s, x_s \mid \mathbf{y}_{\rho_s}, \mathbf{x}_{\eta_s}, \boldsymbol{\psi}) \propto p(y_s \mid \mathbf{y}_{\rho_s}, x_s, \boldsymbol{\psi})\ \ p(x_s \mid \mathbf{x}_{\eta_s}, \boldsymbol{\psi}) \tag{4.10}$$

$$\propto \frac{1}{\Gamma(\mathbf{y}_{\rho_s},\ \mathbf{x}_{\eta_s},\ \boldsymbol{\psi})} \exp\big\{ - U(y_s, x_s \mid \mathbf{y}_{\rho_s},\ \mathbf{x}_{\eta_s},\ \boldsymbol{\psi})\big\} \tag{4.11}$$

where $U(\cdot)$ is the energy function and $\Gamma(\cdot)$ is the normalising function of the conditional distribution. If the parameter vector is divided so that $\boldsymbol{\psi} = [\{\boldsymbol{\phi}_c, c \in \Lambda\}, \boldsymbol{\gamma}]$, where $\boldsymbol{\phi}_c$ corresponds to a vector of model parameters defining the likelihood of observing the pixel value y_s, given its neighbourhood $\mathbf{y}_{\boldsymbol{\rho}_s}$ and label x_s, and $\boldsymbol{\gamma}_c$ corresponds to a vector of hyper-parameters defining the prior on the label lattice $\mathbf{x}$, then the conditional distribution may

be factorised so that

$$p(y_s,\ x_s \mid \mathbf{y}_{\rho_s},\ \mathbf{x}_{\eta_s},\ \boldsymbol{\psi}) \quad \propto \quad p(y_s \mid \mathbf{y}_{\rho_s}, \boldsymbol{\psi}_{x_s})\ \ p(x_s \mid \mathbf{x}_{\eta_s}, \boldsymbol{\gamma}) \tag{4.12}$$

$$\propto \quad \frac{1}{\Gamma(\mathbf{y}_{\rho_s}, \boldsymbol{\psi}_{x_s}, \mathbf{x}_{\eta_s},\ \boldsymbol{\gamma})}\ \ \exp\Big\{\ -U_1(y_s \mid \mathbf{y}_{\rho_s},\ x_s,\ \boldsymbol{\phi}_{x_s}) \\ -\ U_2(x_s \mid \mathbf{x}_{\eta_s},\ \boldsymbol{\gamma})\Big\} \tag{4.13}$$

When considering the entire image the joint distribution takes the form of a Gibbs distribution whose partition function (or normalising function) is usually too complex to evaluate making it infeasible to compare the relative probabilities of two different MRF realisations. An approximation to the Gibbs distribution that allows an absolute probability for an MRF realisation to be calculated is the Pseudo-Likelihood, introduced by Besag[10] and discussed earlier in this thesis (see section 2.5.1, equation 2.60). The Pseudo-Likelihood is simply the product over the complete image $\boldsymbol{\Omega}$, of the full conditionals, each given by equation 4.13:

$$\mathrm{PL}(\mathbf{y}, \mathbf{x} \mid \boldsymbol{\psi}) \quad \triangleq \quad \prod_{s \in \boldsymbol{\Omega}} p(y_s,\ x_s \mid \mathbf{y}_{\rho_s},\ \mathbf{x}_{\eta_s},\ \boldsymbol{\psi}) \tag{4.14}$$

$$= \quad \prod_{s \in \boldsymbol{\Omega}} \frac{1}{\Gamma(\boldsymbol{\phi}_{x_s})}\ \ \exp\left\{-U_1(y_s \mid \mathbf{y}_{\rho_s}, x_s,\ \boldsymbol{\phi}_{x_s})\right\} \tag{4.15}$$

$$\times \quad \frac{\exp\left\{-\sum_{s \in \boldsymbol{\Omega}} U_2(x_s \mid \mathbf{x}_{\eta_s},\ \boldsymbol{\gamma}\right\}}{\prod_{s \in \boldsymbol{\Omega}} \sum_{c \in \boldsymbol{\Lambda}} \exp\left\{-U_2(c \mid \mathbf{x}_{\eta_s},\ \boldsymbol{\gamma}\right\}} \tag{4.16}$$

where $\Gamma(\boldsymbol{\phi}_{x_s})$ is the normalising constant associated with the likelihood for the observed pixel value given its neighbourhood configuration and label.

By applying Bayes' theorem, an approximation to the posterior distribution for the MRF image model may now be formed using the Pseudo-Likelihood. To expand this such that it is also a function of the model order (or number of label classes) k, proper priors must be defined for all the model parameters. The distribution can then be written

$$p(\mathbf{x}, \boldsymbol{\psi}^{(k)}, k \mid \mathbf{y}) \ \approx \ \frac{\mathrm{PL}(\mathbf{y}, \mathbf{x} \mid \boldsymbol{\psi}^{(k)})\ \ p_r(k)\ p_r(\boldsymbol{\gamma}^{(k)})\ \ \prod_{c=0}^{k-1}\ \ p_r(\boldsymbol{\phi}_c)}{\sum_k \int_{\boldsymbol{\psi}^{(k)}} \int_{\mathbf{x}: x_s \in \boldsymbol{\Lambda}^{(k)}} \mathrm{PL}(\mathbf{y}, \mathbf{x} \mid \boldsymbol{\psi}^{(k)})\ \ p_r(k)\ p_r(\boldsymbol{\gamma}^{(k)})\ \ \prod_{c=0}^{k-1}\ \ p_r(\boldsymbol{\phi}_c)} \tag{4.17}$$

where $\boldsymbol{\psi}^{(k)}$ indicates the set of model parameters of the k'th order model, $\boldsymbol{\Lambda}^{(k)}$ is the state space for each element of the hidden data $\mathbf{x}$, and $\boldsymbol{\gamma}^{(k)}$ is the k'th order hyper-parameter vector. Here $p_r(\cdot)$ indicates a prior distribution. It is possible to incorporate various information criteria [66] [96] into the posterior distribution by writing the compensatory

terms into the prior for model order k. However, the results presented in this dissertation were obtained using non-informative or reference priors.

The Isotropic and Gaussian MRF models used as the basis for segmentation algorithms throughout the remainder of this chapter both take Potts models as their prior distribution on the label field $\mathbf{x}$. The differences between the two models occur in their likelihood model for the observed pixel gray-scale data given the label configuration. The principle difference comprises the lack of conditioning on neighbouring pixel gray-scale values in the Isotropic model, thus ensuring its rotational symmetry. To introduce further notation the two models are defined in more detail in the following two subsections, but for a more detailed discussion of the properties of the GMRF, refer to the earlier section 2.2.3.

4.4.1 The Isotropic Markov Random Field Model

The Isotropic MRF model is used to model an image consisting of regions of constant but different gray-scales corrupted with an additive, zero mean, i.i.d. noise process, whose variance may be dependent on the underlying gray-scale value.

For each pixel, the likelihood of its gray-scale value given its underlying label, is given by an Gaussian distribution whose parameters are dependent on the label class. Hence, the gray-scale values of the pixels comprising a region labelled as a single class c may be considered a realisation of an i.i.d. Gaussian noise process whose parameter vector is given by $\boldsymbol{\phi}_c = [\mu_c, \sigma_c]$.

The Potts model chosen to model *a priori* knowledge of spatial correlations within the label field and incorporates potential functions defined using both singular and pairwise cliques on a nearest neighbour type neighbourhood. If the hyper-parameter vector is $\boldsymbol{\gamma} = [\{\alpha_c, c \in \boldsymbol{\Lambda}\}, \beta]$ then the approximation to the posterior density given in equation 4.17 may be written,

$$\begin{aligned} p(\mathbf{x}, \boldsymbol{\psi}^{(k)}, k \mid \mathbf{y}) \;\approx\; & \frac{1}{Z} \prod_{s \in \boldsymbol{\Omega}} \frac{1}{\sqrt{2\pi\sigma_{x_s}^2}} \exp\left\{ -\frac{1}{2\sigma_{x_s}^2} \left(y_s - \mu_{x_s}\right)^2 \right\} \\ & \times \frac{\exp\left\{ -\sum_{s \in \boldsymbol{\Omega}} \left(\alpha_{x_s} + \beta \; V(x_s, \mathbf{x}_{\boldsymbol{\eta}_s}) \right) \right\}}{\prod_{s \in \boldsymbol{\Omega}} \sum_{c \in \boldsymbol{\Lambda}^{(k)}} \exp\left\{ -\left(\alpha_c + \beta \; V(c, \mathbf{x}_{\boldsymbol{\eta}_s}) \right) \right\}} \\ & \times \; p_r(\beta) \;\; p_r(k) \prod_{c \in \boldsymbol{\Lambda}^{(k)}} p_r(\mu_c) \; p_r(\sigma_c) \; p_r(\alpha_c) \end{aligned} \tag{4.18}$$

where Z is the normalising constant between differing models and $V(c, \mathbf{x}_{\boldsymbol{\eta}_s})$ is the potential function at site s when $x_s = c$. Throughout this chapter the potential function is defined,

$V(c, \mathbf{x}_{\boldsymbol{\eta}}) = \frac{1}{4}\sum_{t\in\boldsymbol{\eta}}(c \oplus x_t)$, where $\oplus$ is an operator defined to take the value -1 if its arguments are equal, otherwise $+1$.

Non-informative or reference priors are chosen for the noise model parameters to ensure the observed intensity data dominates any prior information incorporated into the model. Uniform priors are selected for $\{\mu_c, \alpha_c; c \in \mathbf{\Lambda}^{(k)}\}$, β and k. For $\{\sigma_c, \forall c \in \mathbf{\Lambda}^{(k)}\}$ the reference priors may be found using Jeffrey's formula for non-informative priors [8]. Normally these priors are improper, but here their range is restricted to allow normalisation and ensure that the criteria for model selection is valid.

4.4.2 The Gaussian Markov Random Field Model

The Gaussian MRF (GMRF) is commonly used to model images comprising textures characterised by their spatial correlation. The image model used in this paper is hierarchical with individual textures modelled by GMRF's and the interaction between the regions comprising these textures, by a Potts Model (as was used in the Isotropic case).

The c'th GMRF may be parameterised by its covariance matrix $\mathbf{\Sigma}_c$ and mean vector $\boldsymbol{\mu}_c$. If the $\mathbf{y}_c$ is the pixel gray-scale vector and its dimension is N_c, then this leads to a joint distribution,

$$p(\mathbf{y}_c \mid \mu_c, \mathbf{\Sigma}_c) = \frac{\exp\{-\frac{1}{2}\,[\mathbf{y}_c - \boldsymbol{\mu}_c]^T\,[\mathbf{\Sigma}_c^{-1}][\mathbf{y}_c - \boldsymbol{\mu}_c]\}}{(2\pi)^{N_c/2}\;\;|\mathbf{\Sigma}_c|^{\frac{1}{2}}} \tag{4.19}$$

If we assign a neighbourhood to the pixel located at site s and denote it $\boldsymbol{\rho}_s$, then the conditional distribution for its gray-scale value may be written

$$p(y_s \mid \mu_c, \sigma_c, \boldsymbol{\theta}_c, \mathbf{y}_{\boldsymbol{\rho}_s}) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\Big\{ -\frac{1}{2\sigma_c^2}\Big((y_s - \mu_c) - \sum_{\tau: s+\tau\in\rho_s} \theta_c^{(\tau)}\left[(y_{s+\tau} - \mu_c) + (y_{s-\tau} - \mu_c)\right]\Big)^2\Big\} \tag{4.20}$$

where $\boldsymbol{\theta}_c = \{\theta_c^{(\tau)}, \tau \in \rho\}$ is the set of correlation coefficients of the non-causal AR process (see section 2.2.3 for a full explanation). Note also that for simplicity and with no loss of generality we put $\theta_c^{(\tau)} = \theta_c^{(-\tau)}$.

The interaction between regions is again described by a Potts model. As with the Isotropic MRF this implies *a priori* knowledge of spatial correlations within the label field. The Potts model chosen here is identical to that used in the Isotropic case (given in the

previous section) except that there are no single clique parameters. These are omitted to reduce the complexity of the model order sampling step (see section 4.5.2) and to weaken the prior on $\mathbf{x}$. Hence the hyper-parameter vector simply consists of one term, β.

The posterior distribution for the complete hierarchical image model may now be approximated using the expression given in equation 4.17 and the conditional distributions given by equation 4.20:

$$\begin{aligned} p(\mathbf{x}, \boldsymbol{\psi}^{(k)}, k \mid \mathbf{y}) \quad \approx \quad & \frac{1}{Z} \prod_{s \in \boldsymbol{\Omega}} p(y_s \mid \mu_{x_s}, \sigma_{x_s}, \boldsymbol{\theta}_{x_s}) \\ & \times \quad \frac{\exp\left\{-\sum_{s \in \boldsymbol{\Omega}} \beta \; V(x_s, \mathbf{x}_{\boldsymbol{\eta}_s})\right\}}{\prod_{s \in \boldsymbol{\Omega}} \sum_{c \in \boldsymbol{\Lambda}^{(k)}} \exp\left\{-\beta \; V(c, \mathbf{x}_{\boldsymbol{\eta}_s})\right\}} \\ & \times \quad p_r(\beta) \; p_r(k) \prod_{c \in \boldsymbol{\Lambda}^{(k)}} p_r(\mu_c) \; p_r(\sigma_c) \; p_r(\boldsymbol{\theta}_c) \end{aligned} \tag{4.21}$$

where $V(c, \mathbf{x}_{\boldsymbol{\eta}_s})$ is the potential function as defined in the previous section. Priors for all parameters are as defined earlier excepting the $\theta^{(\tau)}$ parameters. These are assigned conjugate priors to reduce the computation given later, in section 4.5.2. These priors therefore consist of a family of normal distributions $N(0, \lambda^2)$, with λ being a common hyper-parameter.

4.5 MCMC Sampling from the Posterior Distribution

The image segmentation, parameter estimates and model order are all estimated to maximise the *a posteriori* probability (known as the MAP criterion) of the model given the observed image. This comprises an optimisation problem over the MRF's label map, parameter space and model order. The approach adopted here was initially developed by Geman and Geman [39]: to construct a Markov Chain to sample from the target distribution and then perform a process of stochastic relaxation (i.e. Simulated Annealing) to find the dominant mode of this distribution.

This philosophy is adopted because the combined model's energy surface can be expected to be highly multi-modal thus deterministic optimisation algorithms are likely to fail (i.e. reach a local minima only). Also when sampling model order (using the reversible

jump algorithms described in sections 3.1 and 3.2) jumps to lower probability areas of the model space occur frequently and thus, to adopt any deterministic elements to the algorithm would appear inconsistent with such a scheme.

Of course by eliminating various shortcuts as reviewed in chapter 2, for example the Iterative Conditional Modes algorithm [11]; using the EM algorithm to estimate model parameters [18], or the MAP based parameter estimation algorithms [96], a great computational burden is incurred. However, the use of *ad hoc* and approximate techniques throughout the literature has meant that currently, no robust solution to the segmentation problem exists. The solution to this problem is therefore the overriding objective to these sections of the dissertation.

The sampling scheme in this chapter is based on the Gibbs Sampler [39] but Metropolis-Hastings sub-chains [93] are incorporated to enable the model parameters and number of classes to be sampled. The sampling process follows a predetermined sequential scan, updating the pixel sites and various model parameters in a specific order. The scan consists of the following sampling processes:

1. re-segment the image,
2. sample noise model parameters,
3. sample hierarchical model hyper-parameters,
4. sample the number of classes.

The first three of these steps are relatively straightforward. All labels and parameters are sampled from their respective conditional distributions. The first step consists of Gibbs sampling the label field. The conditional distributions may be found by applying Bayes' theorem to equation 4.18. The resulting distribution for the Isotropic MRF described in section 2 is given by

$$p(x_s = c \mid y_s, \mathbf{x}_{\boldsymbol{\eta}_s}, \mu_c, \sigma_c, \alpha_c, \beta) \propto \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left\{ -\frac{1}{2}\left(\frac{y_s - \mu_c}{\sigma_c}\right)^2 - \left(\alpha_c + \beta\, V(c, \mathbf{x}_{\boldsymbol{\eta}_s})\right) \right\} \tag{4.22}$$

The distribution for the GMRF is similar,

$$p(x_s = c \mid y_s, \mathbf{y}_{\rho_s}, \mathbf{x}_{\boldsymbol{\eta}_s}, \mu_c, \sigma_c, \boldsymbol{\theta}_c, \beta) \propto \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\Big\{ -\frac{1}{2\sigma_c^2} \Big((y_s - \mu_c) - \sum_{\tau: s+\tau \in \rho_s} \theta_c^{(\tau)} \big[(y_{s+\tau} - \mu_c) + (y_{s-\tau} - \mu_c) \big] \Big)^2 - \beta\, V(c, \mathbf{x}_{\boldsymbol{\eta}_s}) \Big\} \tag{4.23}$$

Metropolis-Hastings sampling is used to update noise and MRF model parameters. The proposal densities used are zero mean Gaussian with variances dependent on the parameter being sampled. The conditional distributions for the noise model parameters are found by multiplying the appropriate pseudo-likelihood terms by the model parameter priors. As specified in sections 2.1 and 2.2, for both models, non-informative priors are used for the μ and σ parameters. However, conjugate priors are used for the $\theta^{(\tau)}$ parameters of the GMRF. The resulting conditional distribution for the Isotropic case is

$$\begin{aligned} p(\mu_c, \sigma_c \mid \mathbf{y},\ \mathbf{x}) &\propto \prod_{s: x_s = c} p(y_s \mid \mu_c, \sigma_c)\ p_r(\mu_c) p_r(\sigma_c) \\ &\propto \frac{1}{\sigma_c\ (2\pi\sigma_c^2)^{N_c}} \exp\left\{ -\frac{1}{2} \sum_{s: x_s = c} \left(\frac{y_s - \mu_c}{\sigma_c} \right)^2 \right\} \end{aligned} \tag{4.24}$$

where N_c represents the number of sites allocated to class c, over the hidden label field $\mathbf{x}$. For the c'th GMRF texture model, the model parameter conditional distribution is given by

$$\begin{aligned} &p(\mu_c, \sigma_c, \boldsymbol{\theta}_c \mid \mathbf{y},\ \mathbf{x}) \\ &\quad\propto \prod_{s: x_s = c} p(y_s \mid \mu_c, \sigma_c, \boldsymbol{\theta}_c)\ p_r(\mu_c) p_r(\sigma_c) p_r(\boldsymbol{\theta}_c) \\ &\quad\propto \frac{1}{\sigma_c\ (2\pi\sigma_c^2)^{n_c}} \exp\Big\{ -\frac{1}{2\sigma_c^2} \sum_{s: x_s = c} \Big((y_s - \mu_c) - \sum_{\tau: s+\tau \in \boldsymbol{\rho}_s} \theta_c^{(\tau)} \big[(y_{s+\tau} - \mu_c) + (y_{s-\tau} - \mu_c) \big] \Big)^2 \Big\} \end{aligned} \tag{4.25}$$

To sample the hyper-parameters for the prior on the label parameters, the Metropolis-Hastings algorithm is used to sample an approximation to the full conditional distribution. The approximation used is again the pseudo-likelihood.

The conditional distributions for the external field parameters $\{\alpha_c, c \in \Lambda\}$ used in the

Isotropic model take uniform priors and are given by

$$p(\alpha_c, c \in \mathbf{\Lambda}^{(k)} \mid \mathbf{x}) \quad \propto \quad p(\mathbf{x} \mid \alpha_c, c \in \mathbf{\Lambda}^{(k)})\ p_r(\alpha_c, c \in \mathbf{\Lambda}^{(k)}) \tag{4.26}$$

$$\propto \quad \prod_{(c \in \mathbf{\Lambda}^{(k)},\ \forall \mathbf{x}_{\boldsymbol{\eta}})} \left(\frac{\exp\left(-[\alpha_c + \beta\ V(c, \mathbf{x}_{\boldsymbol{\eta}})]\right)}{\sum_{i \in \mathbf{\Lambda}} \exp\left(-[\alpha_i + \beta\ V(i, \mathbf{x}_{\boldsymbol{\eta}})]\right)} \right)^{N_{(c, \mathbf{x}_{\boldsymbol{\eta}})}} \tag{4.27}$$

where $N_{(c,\mathbf{x}_\eta)} = \#(s : x_s = c, \mathbf{x}_{\eta_s} = \mathbf{x}_\eta)$ and $\#(\cdot)$ is the operator, 'the number of', and the product is over all states and possible neighbourhood configurations.

It is theoretically possible to sample the β parameter from its conditional distribution,

$$\begin{aligned} p(\beta \mid \mathbf{x}) \ &\propto\ p(\mathbf{x} \mid \beta)\ p_r(\beta) \\ &\propto \prod_{(c \in \mathbf{\Lambda}, \forall \mathbf{x}_{\boldsymbol{\eta}})} \left(\frac{\exp\left(-\beta\ V(c, \mathbf{x}_{\boldsymbol{\eta}})\right)}{\sum_{i \in \mathbf{\Lambda}} \exp\left(-[\alpha_i + \beta\ V(i, \mathbf{x}_{\boldsymbol{\eta}})]\right)} \right)^{N_{(c, \mathbf{x}_{\boldsymbol{\eta}})}} \end{aligned} \tag{4.28}$$

Unfortunately, under particular underlying label map configurations, for example $\mathbf{x}'$, the posterior distribution for β will be improper.

$$\lim_{\beta \to -\infty} p(\mathbf{X} = \mathbf{x}', \psi^{(k)} \mid \mathbf{y}) \ = \ \delta \tag{4.29}$$

where δ is a positive constant. This follows as a direct consequence of approximating the likelihood by the pseudo-likelihood (see section 2.5.1). It may be possible to overcome this problem by choosing a suitable prior for β. However, for simplicity the results presented in section 4.6 of this paper are obtained with β set *a priori*.

To sample the model order reversible jumps are incorporated into the Markov Chain. Reversible jumps were developed by Green [43] to allow a Metropolis-Hastings based algorithm to sample the model order. A full description of the Reversible Jump MCMC technique was given in section 4.3. Hence, in the following two subsections, a description is given of how reversible jump algorithms may be used to allow the size of the label space (model order) to be sampled for the Isotropic and Gaussian MRF models.

4.5.1 Reversible Jumps for the Isotropic MRF

The approach to sampling the model of an Isotropic Markov Random Field adopted here is similar to that adopted by Richardson and Green [82] when sampling the number of components of a mixture distribution. As discussed in the previous section, the problem revolves around proposing to split a region labelled by a single class c into a region composed of two classes, c_1 and c_2.

Each class is defined by a parameter vector $\boldsymbol{\psi}_c = \{\mu_c, \sigma_c, \alpha_c\}$. When splitting one class into two, three new parameters need to be created. To ensure the model spaces remain of equal dimension, the parameter space of the original model is supplemented by three random variables e_1, e_2, e_3. Thus when splitting state c into c_1 and c_2 a transform between $[\mu_c, \sigma_c, \alpha_c]$ and $[\mu_{c1}, \sigma_{c1}, \alpha_{c1}, \mu_{c2}, \sigma_{c2}, \alpha_{c2}]$ may be defined with three degrees of freedom.

The three parameters for each of the two new classes are derived from the original parameter values $\mu_c, \sigma_c, \alpha_c$, and the random variables e_1, e_2, e_3. The new parameters are calculated to preserve the 0th, 1st and 2nd order moments across the transformation. The resulting mapping functions are given by the following set of equations:

$$\begin{aligned}
\alpha_{c1} &= \alpha_c - ln(e_1) & \alpha_{c2} &= \alpha_c - ln(1-e_1) \\
\mu_{c1} &= \mu_c - e_2\ \sigma_c \sqrt{\tfrac{1-e_1}{e_1}} & \mu_{c2} &= \mu_c + e_2\ \sigma_c \sqrt{\tfrac{e_1}{1-e_1}} \\
\sigma^2_{c1} &= e_3\ (1-e_2^2)\ \sigma_c^2\ \tfrac{1}{e_1} & \sigma^2_{c2} &= (1-e_3)\ (1-e_2^2)\ \sigma_c^2\ \tfrac{1}{1-e_1}
\end{aligned} \tag{4.30}$$

The choice of random variables for e_1, e_2, e_3, must be such that $\{e_i \in (0,1], i = 1,2,3\}$. For this reason and to allow a bias towards splitting the data into roughly equal partitions, beta distributions are used to propose e_1, e_2, e_3.

The Jacobian determinant of these mapping functions needed in the calculation of the jump algorithm's acceptance ratio is given by

$$\left| \frac{\partial(\boldsymbol{\psi}_{c1}, \boldsymbol{\psi}_{c2})}{\partial(\boldsymbol{\psi}_c, e_1, e_2, e_3)} \right| = \frac{\sigma_c^2}{e_1^2(1-e_1^2)\ \sqrt{e_3(1-e_3)}} \tag{4.31}$$

The pixel sites labelled by the class or classes selected to be split or merged must be reallocated on the basis of the new parameters generated. If a merge is being proposed, then all sites allocated to the two old classes are relabelled by the new merged class label, with probability one.

The difficulty occurs when splitting one class into two. If a reasonable probability of acceptance is to be maintained, the proposed re-allocation of labels to sites needs to be completed in such a way as to ensure the posterior probability of that particular segmentation is relatively high. To achieve this it would be desirable to propose a re-allocation of labels by Gibbs sampling from the conditional distributions given by equation 4.22. Unfortunately this is not possible since the allocation of classes in the neighbourhood $\boldsymbol{\eta}_s$ on which the probabilities will be conditioned is not available.

To overcome this problem a deterministic estimate is made of the future allocation of each pixel labelled by the class to be split. The estimate is made at each pixel site using a distance measure between the model distribution functions of the new classes and the histogram of observed gray-scale values of the surrounding region of pixels. The measure used here has a precedent in this type of algorithm: the Kolmogorov-Smirnov distance was used by Geman *et al* [37] to allocate pixel sites between classes based on gray-scale values or particular transformations of gray-scale, indicative of texture type (see section 3.3.1).

The Kolmogorov-Smirnov distance is a measure of the closeness of two distribution functions. It may be applied to two samples of data to ascertain whether they have been drawn independently from the same distribution. If $\hat{F}_1(\kappa)$ and $\hat{F}_2(\kappa)$ are two independent sample distribution functions (i.e. histograms) defined

$$\hat{F}(\kappa) \;=\; \frac{1}{n}\,\#(i : y_i \leq \kappa) \tag{4.32}$$

where n is the number of data samples so that $1 \leq i \leq n$, then the Kolmogorov-Smirnov distance is the maximum difference between distributions over all κ:

$$d(y^{(1)}, y^{(2)}) \;=\; \max_{\kappa} \left| \hat{F}_1(\kappa) - \hat{F}_2(\kappa) \right| \tag{4.33}$$

The Kolmogorov-Smirnov distance is useful for two reasons: its value is independent of the underlying distribution function and it is relatively unaffected by outlying data values. Hence, it may be expected to closely model the label field of an Isotropic MRF, as was found by Geman *et al* [37].

The deterministic estimation of the new label allocation may now be used as the basis for the Gibbs sampling of pixels from their full conditional distributions, a further step necessary to ensure the reversibility of the process when considering the opposite, merging process.

The acceptance ratio for splitting region c into $c1$ and $c2$, thus increasing the number of classes from k to $k+1$, may now be given by,

$$\begin{aligned} &\frac{p(\mathbf{x}', \boldsymbol{\psi}^{(k+1)}, k+1 \mid \mathbf{y})}{p(\mathbf{x}, \boldsymbol{\psi}^{(k)}, k \mid \mathbf{y})} \;\; \frac{q(\text{merge})}{q(\text{split})} \;\; \frac{1}{q_\beta(u_1) q_\beta(u_2) q_\beta(u_3)} \\ &\qquad \times \;\; \frac{1}{q(re-segmentation)} \;\; \left| \frac{\partial(\boldsymbol{\psi}_{c1}, \boldsymbol{\psi}_{c2})}{\partial(\boldsymbol{\psi}_c, u_1, u_2, u_3)} \right| \end{aligned} \tag{4.34}$$

where: $p(\mathbf{x}, \boldsymbol{\psi}^{(k)}, k \mid \mathbf{y})$ is the approximation to the posterior density defined by equation 4.18; $q(\text{merge})$ and $q(\text{split})$ are the probabilities of proposing to merge or split a state; $q_\beta(u)$

is the probability of proposing the random variable u drawn from a beta distribution; $q(re-segmentation)$ is the probability of the re-allocation of the pixels labelled by c into regions labelled by $c1$ and $c2$; the final term is the Jacobian determinant given by equation 4.31. The probabilities of choosing states to split or merge are identical and are thus eliminated from the acceptance ratio. The acceptance ratio for the jump combining two states into one is simply the inverse of that given in equation 4.34. In this case $q(re-segmentation)$ is found in retrospect and u_1, u_2, u_3 are calculated by back-substitution into equation 4.30.

To summarise, when splitting one class of the hierarchical image model into two, the corresponding iteration of the reversible jump algorithm will consist of the following steps,

---

1. randomly select with probability $\frac{1}{k}$ a state from $\mathbf{\Lambda}^{(k)}$ to split,
2. generate the two new parameter vectors $\boldsymbol{\psi}_{c1}, \boldsymbol{\psi}_{c2}$ by drawing the random vector $\mathbf{e}$ from the respective proposal distribution and substituting, together with the old parameter vector $\boldsymbol{\psi}_c$ in the mapping functions contained in equation 4.30,
3. calculate the transition probability $q(\mathbf{e})$ using the values obtained in step(1),
4. calculate the Radon-Nikodym Derivative $\left|\frac{\partial(\boldsymbol{\psi}_{c1},\boldsymbol{\psi}_{c2})}{\partial(\boldsymbol{\psi}_c,u_1,u_2,u_3)}\right|$ substituting the values of step(1),
5. obtain deterministically, using the specified algorithm and Kolmogorov-Smirnov statistic an initial re-segmentation into the two new classes $c1$ and $c2$ of all data previously labelled to state c,
6. conditioning on the initial estimate, use Gibbs sampling to obtain a reversible re-allocation $c \rightarrow c1, c2$, with probability $q(re-segmentation)$,
7. calculate the overall acceptance ratio according to equation 4.34 and accept or reject the proposal based upon this value.

When considering a merge, the same process is followed but applying the opposite transitions or calculations at each step. Thus, if merging classes $c1, c2$ into c, the probability of achieving the reverse split must be calculated and substituted into the inverted acceptance ratio of equation 4.34. When choosing classes to merge, a single class is selected at random, then a second is chosen whose μ parameter is above and closest in value to that of the first.

4.5.2 Reversible Jumps for the Gaussian MRF

Proposing a move when using a reversible jump to sample the model order of an hierarchical GMRF is more complex than generating a similar proposal when sampling the Isotropic MRF for the simple reason, the model parameter vector $\boldsymbol{\phi}_c = [\mu_c, \sigma_c, \boldsymbol{\theta}_c]$ is longer. When splitting a GMRF class, two new GMRF parameter vectors need to be generated. The number of degrees of freedom for the generation of the new vectors is therefore dependent on the size of the old parameter vector and in particular, upon the size of the GMRF neighbourhood. If the dimension of the correlation coefficient vector associated with this neighbourhood is denoted N_θ, then the number of random variables needed to equalise the two model parameter vectors will be $N_\theta + 2$ which is equivalent to the number of degrees of freedom in generating the new model parameters.

If a large degree of freedom is allowed in generating a large number of parameters then the likelihood of proposing a reversible jump that has a reasonable chance of being accepted will be small and so convergence of the Markov Chain will be slow. To alleviate this problem the model order is sampled from a marginal density and the method of composition sampling [92] is then employed to obtain the remaining parameters. Composition sampling requires the availability of factorisations of the posterior density. The marginal we propose to use eliminates all but one correlation parameter from the posterior density:

$$p(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{x} \mid \mathbf{y}) \;=\; p(\boldsymbol{\theta}^{(-i)} \mid \boldsymbol{\theta}^{(i)}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{x}, \mathbf{y})\; p(\boldsymbol{\theta}^{(i)}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{x} \mid \mathbf{y}) \tag{4.35}$$

where, $\boldsymbol{\theta}^{(-i)} = [\boldsymbol{\theta}_c^{(-i)}; c \in \Lambda]$ and $\boldsymbol{\theta}_c^{(-i)} = [\theta_c^{(1)}, .., \theta_c^{(i-1)}, \theta_c^{(i+1)}, .., \theta_c^{(N_{AR})}]$. Similarly $\boldsymbol{\theta}^{(i)} = [\boldsymbol{\theta}_c^{(i)}; c \in \Lambda]$, $\boldsymbol{\mu} = [\mu_c; c \in \Lambda]$ and $\boldsymbol{\sigma} = [\sigma_c; c \in \Lambda]$.

The above factorisation of the posterior density can only be achieved by incorporating the pseudo-likelihood approximation into the marginal. Applying Bayes' theorem, the marginal may be expressed,

$$p(\boldsymbol{\theta}^{(i)}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{x} \mid \mathbf{y}) \;\propto\; \mathrm{PL}(\mathbf{y} \mid \lambda, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\theta}^{(i)}, \mathbf{x}) \left(\prod_{s \in \boldsymbol{\Omega}} p(x_s \mid \mathbf{x}_{\boldsymbol{\eta}_s}) \right) p_r(\boldsymbol{\theta}^{(i)}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \tag{4.36}$$

All the terms in the above equation are easily defined except the marginal of the pseudo-likelihood $\mathrm{PL}(\mathbf{y} \mid \lambda, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\theta}^{(i)}, \mathbf{x})$. This is found by first constructing the full pseudo-likelihood and multiplying it by priors on all the parameters to be eliminated on forming the marginal. These parameters are then integrated out of the resulting expression to form that marginal.

To facilitate the analytical integration over the correlation parameter space, conjugate priors are used which take the form of Normal distributions $N(0, \lambda^2)$. The pseudo-likelihood is therefore marginalised by evaluating the following product of integrals

$$\begin{aligned} \text{PL} \quad & (\mathbf{y} \mid \lambda, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\theta}^{(i)}, \mathbf{x}) \qquad (4.37) \\ & \triangleq \prod_{c \in \mathbf{\Lambda}} \int_{\Theta_c^{(-i)}} \Big(\prod_{s:x_s=c} p(y_s \mid \mu_c, \sigma_c, \boldsymbol{\theta}_c, \mathbf{y}_{\eta_s}) \Big) \; p(\boldsymbol{\theta}_c^{(-i)}) \;\; d\boldsymbol{\theta}_c^{(-i)} \\ & = \prod_{c \in \mathbf{\Lambda}} \frac{(2\pi\sigma_c^2 t)^{\frac{-N_c+N_\theta-1}{2}}}{|\mathbf{D}_c|^{\frac{1}{2}}} \; (2\pi\lambda^2 t)^{\frac{-N_\theta+1}{2}} \; \exp\left\{ -\frac{1}{2\sigma_c^2 t} \; \big(h(\theta_c^{(i)} \mid \mu_c, \lambda, \mathbf{y}) \big) \right\} \end{aligned}$$

where

$$\begin{aligned} h(\theta_c^{(i)} \mid \mu_c, \mathbf{y}) &= -\mathbf{v}^T [\mathbf{D}_c]^{-1} \mathbf{v} + \sum_{s:x_s=c} \Big(y'_s - \theta_c^{(i)} [y'_{s+\tau_i} + y'_{s-\tau_i}] \Big)^2 && (4.38) \\ \mathbf{v} &= \Big[\sum_{s:x_s=c} (y'_s - \theta_c^{(i)} [y'_{s+\tau_i} + y'_{s-\tau_i}] \; \mathbf{y}_{(\boldsymbol{\rho}\backslash i)_s}) \Big] && (4.39) \\ \mathbf{D}_c &= \Big[\lambda \mathbf{I} + \sum_{s:x_s=c} \mathbf{y}^T_{(\boldsymbol{\rho}\backslash i)_s} \mathbf{y}_{(\boldsymbol{\rho}\backslash i)_s} \Big] && (4.40) \\ \mathbf{y}^T_{(\boldsymbol{\rho}\backslash i)_s} &= \Big[(y'_{s+\tau_1} + y'_{s-\tau_1}), .., (y'_{s+\tau_{i-1}} + y'_{s-\tau_{i-1}}), \\ & \qquad (y'_{s+\tau_{i+1}} + y'_{s-\tau_{i+1}}), .., (y'_{s+\tau_{N_\theta}} + y'_{s-\tau_{N_\theta}}) \Big] && (4.41) \\ y'_s &= y_s - \mu_c && (4.42) \end{aligned}$$

We now have an expression for the marginal likelihood which when multiplied by the relevant priors gives the basis for the reversible jump sampling of the model order. To complete the algorithm, methods to propose parameters for the new classes and to propose new segmentations for the marginalised GMRF are required.

To allow a one-to-one mapping between old and new parameter vectors random variables are introduced. Transforms need to be defined for both the splitting of one state into two and the inverse, the combining of two into one. The parameter vector for a single state, augmented by the additional random variables is $[\mu_c, \sigma_c, \theta_c^{(i)}, e_{m_m}, e_{m_e}, e_{s_m}, e_{s_e}, e_{t_m}, e_{t_e}]$. This may be transformed to give the two new parameter vector and some further random variables, $[\mu_{c_1}, \sigma_{c_1}, \theta_{c_1}^{(i)}, \mu_{c_2}, \sigma_{c_2}, \theta_{c_2}^{(i)}, a_m, a_s, a_t]$. In the above vectors
$e_{m_m}, e_{m_e}, e_{s_m}, e_{s_e}, e_{t_m}, e_{t_e}, a_m, a_s, a_t$ are drawn from a combination of zero mean Gaussian distributions with differing variances, and shifted Gamma distributions, thus ensuring the positivity of proposed variance parameters. The split move is defined by the transforma-

tions:

$$\begin{aligned}
\mu_{c_1} &= \mu_c - e_{m_m} & \mu_{c_2} &= \mu_c + e_{m_m} + e_{m_e} \\
\sigma_{c_1} &= \sigma_c - e_{s_m} & \sigma_{c_2} &= \sigma_c + e_{s_m} + e_{s_e} \\
\theta_{c_1}^{(i)} &= \theta_c^{(i)} - e_{t_m} & \theta_{c_1}^{(i)} &= \theta_c^{(i)} + e_{t_m} + e_{t_e}
\end{aligned} \tag{4.43}$$

The combine move is designed to preserve the 1st and 2nd order central moments in a similar way to that used in the Isotropic case and by Richardson and Green [82] when sampling mixture distributions. In addition, a small perturbation in the parameters is allowed, given by the random variables a_m, a_s, a_w. The resulting transforms are:

$$\begin{aligned}
\mu_c &= \tfrac{1}{N_c}\Big[\mu_{c_1} N_{c_1} + \mu_{c_2} N_{c_2}\Big] \; + \; a_m \\
\sigma_c &= \tfrac{1}{N_c}\Big[(\sigma_{c_1}^2 + \mu_{c_1}^2) N_{c_1} + (\sigma_{c_2}^2 + \mu_{c_2}^2) N_{c_2}\Big] - \mu_c^2 - a_s \\
\theta_c^{(i)} &= \tfrac{1}{N_c}\Big[\theta_{c_1}^{(i)} N_{c_1} + \theta_{c_2}^{(i)} N_{c_2}\Big] \; + \; a_t
\end{aligned} \tag{4.44}$$

where N_c is the number of pixels assigned to class c, N_{c_1} to class c_1 and N_{c_2} to class c_2.

The accepted reversible jump framework [43] [82] for proposing a split is to begin by generating new parameters for the two new states, then Gibbs sample the data allocated to the split state into each of the two new states based on their full conditionals. In this case, new parameters may be generated in the traditional manner. However, Gibbs sampling of the data labels is impossible because of the non-causal nature of the GMRF. The conditionals are dependent on data labels yet to be allocated.

To overcome this problem, a refinement of the methodology introduced in the previous section for the Isotropic MRF reversible jump is used: a likely re-allocation of the label field is first estimated in a deterministic fashion; this is then used to condition the full conditional distributions for the label parameters to allow a Gibbs sampling sweep of all sites labelled with the class being split.

To estimate a likely label field configuration for the GMRF, a square window, denoted W_s is considered around each pixel in the image. The pseudo-likelihoods of allocating all

the pixels in the window to each of the two new classes are calculated and the pixel is allocated to the class with the larger pseudo-likelihood.

The pseudo-likelihood for each window is calculated as in equation 4.37, except the product is over all pixels in the window rather than all pixels labelled by state. Hence, the pseudo-likelihood of the pixel at site s being labelled by class c, given its surrounding window is

$$\begin{aligned}&\mathrm{PL}_c(\{y_r; r \in W_s\} \mid \lambda, \mu_c, \sigma_c, \theta_c^{(i)}, \{\mathbf{y}_{\eta_r}; r \in W_s\}) \\ &\quad \triangleq \left(\frac{1}{2\pi\sigma_c^2 t}\right)^{\frac{N_{W_s}-(N_\theta-1)^2)}{2}} \frac{1}{|\mathbf{D}_{W_s}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_c^2 t}\ \left(h_w(\theta_c^{(i)} \mid \mu_c, \mathbf{y})\right)\right\}\end{aligned}$$

where

$$h(\theta_c^{(i)} \mid \mu_c, \mathbf{y}) = -\mathbf{v}^T[\mathbf{D}_{W_s}]^{-1}\mathbf{v} + \sum_{r \in W_s}\left(y'_r - \theta_c^{(i)}[y'_{r+\tau_i} + y'_{r-\tau_i}]\right)^2 \tag{4.45}$$

$$\mathbf{v} = \left[\sum_{r \in W_s}(y'_r - \theta_c^{(i)}[y'_{r+\tau_i} + y'_{r-\tau_i}]\ \mathbf{y}_{(\boldsymbol{\rho}\setminus i)_r})\right] \tag{4.46}$$

$$\mathbf{D}_{W_s} = \left[\lambda\mathbf{I} + \sum_{r \in W_s}\mathbf{y}^T_{(\boldsymbol{\rho}\setminus i)_r}\mathbf{y}_{(\boldsymbol{\rho}\setminus i)_r}\right] \tag{4.47}$$

and N_{W_s} is the size of the window surrounding pixel s. $\mathbf{y}_{(\boldsymbol{\rho}\setminus i)_r}$ and y'_r are defined as in equations 4.41 and 4.42.

Using this estimation of the new label field, denoted $\mathbf{x}^{(e)}$, the proposal label field can be Gibbs sampled using the marginalised conditional distributions, thus ensuring the reversibility condition of the reversible jump algorithm is met (i.e. the proposition of the reverse move, the merger from the new split segmentation to the single state, is possible). The conditional distributions are given by

$$\begin{aligned}p(x_s = &c \mid \lambda, \mu_c, \sigma_c, y_s, \mathbf{y}_{\rho_s}, \mathbf{x}^{(e)}_{\boldsymbol{\eta}_s}) \\ &\propto p(y_s \mid \lambda, \mu_c, \sigma_c, \theta_c^{(i)}, \mathbf{y}_{\rho_s})\ p(x_s = c \mid \beta^{(1)}, \mathbf{x}^{(e)}_{\boldsymbol{\eta}_s})\end{aligned} \tag{4.48}$$

where $\boldsymbol{\eta}_s$ is the neighbourhood of interactions within the label field for site s. The marginalised likelihood term in this equation may be evaluated:

$$\begin{aligned}p(y_s \mid \lambda, &\ \mu_c, \sigma_c, \theta_c^{(i)}, \mathbf{y}_{\boldsymbol{\rho}_s}) \\ &= \int_{\boldsymbol{\Theta}_{-i}} p(y_s \mid \mu_c, \sigma_c, \boldsymbol{\theta}_c, \mathbf{y}_{\boldsymbol{\rho}_s}) p(\boldsymbol{\theta}^{(-i)})\ \ d\boldsymbol{\theta}^{(-i)} \\ &= \frac{1}{(2\pi\sigma_c^2)^{\frac{(N_\theta-1)^2-1}{2}}\ |\mathbf{D}_s|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_c^2}\ \left(h_s(\theta^{(i)} \mid \mu_c, \mathbf{y}_{\boldsymbol{\rho}_s})\right)\right\}\end{aligned} \tag{4.49}$$

where,

$$\begin{aligned} h_s(\theta^{(i)} \mid \mu_c, \mathbf{y}_{\boldsymbol{\eta}_s}) &= \left[y'_s - \theta^{(i)}[y'_{s+\tau_i} + y'_{s-\tau_i}]\right]^2 \left[1 - \mathbf{y}^T_{(\boldsymbol{\rho}\backslash i)_s}[\mathbf{D}_s]^{-1}\mathbf{y}_{(\boldsymbol{\rho}\backslash i)_s}\right] \\ \mathbf{D}_s &= [\mathbf{y}^T_{(\boldsymbol{\rho}\backslash i)_s}\mathbf{y}_{(\boldsymbol{\rho}\backslash i)_s} + \lambda\mathbf{I}] \end{aligned} \tag{4.50}$$

$\mathbf{y}_{(\boldsymbol{\rho}\backslash i)_s}$ and y'_s are as defined in equations 4.41 and 4.42.

The prior on $\mathbf{x}$ comprises an Ising model, hence the conditional for x_s, given its vector of neighbourhood labels $\mathbf{x}^{(e)}_{\boldsymbol{\eta}_s}$, is a conditional Gibbs distribution:

$$p(x_s = c \mid \beta, \mathbf{x}^{(e)}_{\boldsymbol{\eta}_s}) = \frac{\exp\left\{-\beta\left[\delta\ V_f(x_s = c \mid \mathbf{x}^{(e)}_{\boldsymbol{\eta}_s}) + V_b(x_s = c \mid \mathbf{x}^{(e)}_{\boldsymbol{\eta}_s})\right]\right\}}{\sum_{x\in\boldsymbol{\Lambda}} \exp\left\{-\beta\left[\delta\ V_f(x_s = x \mid \mathbf{x}^{(e)}_{\boldsymbol{\eta}_s}) + V_b(x_s = \mathcal{X} \mid \mathbf{x}^{(e)}_{\boldsymbol{\eta}_s})\right]\right\}} \tag{4.51}$$

The acceptance ratio for the resulting segmentation and parameter estimates for splitting region c into $c1$ and $c2$ is given by min[1,R], where

$$\begin{aligned} R = {} & \frac{p^{(-i)}(\mathbf{x}', \boldsymbol{\psi}^{(k+1)}, k+1 \mid \mathbf{y})}{p^{(-i)}(\mathbf{x}, \boldsymbol{\psi}^{(k)}, k \mid \mathbf{y})} \frac{q(\text{merge})}{q(\text{split})} \frac{1}{k+1} \frac{q(a_m)q(a_s)q(a_t)}{q(u_{m_m})q(u_{m_e})q(u_{s_m})q(u_{s_e})q(u_{t_m})q(u_{t_e})} \\ & \times \frac{1}{q(re-segmentation)} \left|\frac{\partial(\boldsymbol{\phi}^{(-i)}_{c1}, \boldsymbol{\phi}^{(-i)}_{c2}, a_m, a_s, a_t)}{\partial(\boldsymbol{\phi}^{(-i)}_{c}, u_{m_m}, u_{m_e}, u_{s_m}, u_{s_e}, u_{t_m}, u_{t_e})}\right| \end{aligned} \tag{4.52}$$

and where: $p^{(-i)}(\mathbf{x}, \boldsymbol{\psi}^{(k)}, k \mid \mathbf{y})$ is the approximation to the marginalised posterior distribution defined by equations 4.36 to 4.42, with $\boldsymbol{\phi}^{(-i)}_c$ denoting the reduced parameter vector of the marginal likelihood function for class c; $\frac{q(\text{merge})}{q(\text{split})}$ is the ratio of the proposal probabilities of the merging or splitting processes; $q(e)$ is the probability of proposing the random variable u drawn from a normal distribution; $q(a)$ is the retrospective probability of drawing the random variable a from another normal distribution; $q(re-segmentation)$ is the probability of the re-allocation of the pixels labelled by c into regions labelled by $c1$ and $c2$, and the final term is the Jacobian determinant given by equation 4.31. If the proposed split (or merge) is accepted, then the remaining θ parameters are repeatedly sampled from their full conditional distributions given by equation 4.25.

To summarise, the reversible jump algorithm, when splitting one class of the hierarchical GMRF image model into two will consist of the following steps,

---

1. randomly select a class from $\boldsymbol{\Lambda}^{(k)}$ to split, with probability $\frac{1}{k}$,

2. randomly select the i'th GMRF correlation parameter $\theta^{(i)}$, to form the basis for the split by drawing from a uniform distribution, with probability $\frac{1}{N_{AR}}$,

3. draw the random vector $[e_{m_m}, e_{m_e}, e_{s_m}, e_{s_e}, e_{t_m}, e_{t_e}]$ from the respective proposal distributions with probability $q(\mathbf{e})$, and substitute, together with the old marginal parameter vector $\boldsymbol{\phi}_c^{(-i)} = [\mu_c, \sigma_c, \theta_c^{(i)}]$ into equations 4.43 to form two new parameter vectors $\boldsymbol{\phi}_{c_1}^{(-i)} = [\mu_{c_1}, \sigma_{c_1}, \theta_{c_1}^{(i)}]$ and $\boldsymbol{\phi}_{c_2}^{(-i)} = [\mu_{c_2}, \sigma_{c_2}, \theta_{c_2}^{(i)}]$.

4. calculate the probability of generating the merged parameter vector $\boldsymbol{\phi}_c^{(-i)}$ as if generating the reverse or merging proposal according to equations 4.44. This will correspond to the probability of generating the random vector $[a_m, a_s, a_t]$ as calculated by substituting the parameter values obtained in the previous step into equations 4.44.

5. calculate the Radon-Nikodym Derivative $\left| \frac{\partial\left(\boldsymbol{\phi}_{c_1}^{(-i)}, \boldsymbol{\phi}_{c_2}^{(-i)}\right)}{\partial\left(\boldsymbol{\phi}_c^{(-i)}, e_1, e_2, e_3\right)} \right|$ substituting the values of step(3),

6. obtain deterministically, using the specified algorithm and Pseudo-Likelihood statistics an initial re-segmentation into the two new classes c_1 and c_2, of all data previously labelled to state c,

7. conditioning on the initial estimate, use Gibbs sampling to obtain a reversible re-allocation $c \rightarrow c_1, c_2$, with probability $q(re-segmentation)$,

8. calculate the overall acceptance ratio according to equation 4.34 and accept or reject the proposal based upon this value,

9. should the proposal be accepted, repeatedly sample the unknown parameters $\boldsymbol{\theta}_{c_1}^{(-i)}$ and $\boldsymbol{\theta}_{c_2}^{(-i)}$ from their full conditional distributions, given by equation 4.25.

When considering a merge, the similar process is followed but applying the opposite transitions or calculations at each step. Thus, if merging classes $c1$ and $c2$ into c, the probability of achieving the reverse split must be calculated and substituted into the inverted acceptance ratio of equation 4.34. In summary,

---

1. randomly select a pair of classes from $\boldsymbol{\Lambda}^{(k)}$ to be split, with probability $\frac{1}{k(k-1)}$,

2. randomly select the i'th GMRF correlation parameter $\theta^{(i)}$, to form the basis for the split by drawing from a uniform distribution, with probability $\frac{1}{N_{AR}}$,

3. draw the random vector $[a_m, a_s, a_w]$ from the respective proposal distributions with probability $q(\mathbf{a})$, and substitute, together with the pair of marginal parameter vectors $\boldsymbol{\phi}_{c_1}^{(-i)} = [\mu_{c_1}, \sigma_{c_1}, \theta_{c_1}^{(i)}]$ and $\boldsymbol{\phi}_{c_2}^{(-i)} = [\mu_{c_2}, \sigma_{c_2}, \theta_{c_2}^{(i)}]$, into equations 4.44 to form the merged proposal parameter vector $\boldsymbol{\phi}_{c}^{(-i)} = [\mu_c, \sigma_c, \theta_c^{(i)}]$,

4. calculate the probability of generating the parameter vectors $\boldsymbol{\phi}_{c_1}^{(-i)}$ and $\boldsymbol{\phi}_{c_2}^{(-i)}$ as if proposing the split $c \rightarrow c_1, c_2$, according to equations 4.43. This will correspond to the probability of generating the random vector $[e_{m_m}, e_{m_e}, e_{s_m}, e_{s_e}, e_{t_m}, e_{t_e}]$ as calculated by substituting the parameter values obtained in the previous step into equations 4.43,

5. calculate the Radon-Nikodym Derivative $\left| \frac{\partial\left(\boldsymbol{\phi}_c^{(-i)}, e_1, e_2, e_3\right)}{\partial\left(\boldsymbol{\phi}_{c1}^{(-i)}, \boldsymbol{\phi}_{c2}^{(-i)}\right)} \right|$ by substituting the values obtained in step (3),

6. obtain deterministically, by using the algorithm based upon Pseudo-Likelihood statistics deterministic the Gibbs sampler probability of proposing the reverse re-allocation of region c into the two new regions c_1 and c_2,

7. calculate the overall acceptance ratio according to the inverse of equation 4.34 and accept or reject the merge based upon this value.

8. should the proposal be accepted, repeatedly sample the unknown parameters $\boldsymbol{\theta}_c^{(-i)}$ from their full conditional distributions, given by equation 4.25.

4.6 Experimental Results

There has been much debate on the subject of how convergence might relate to annealing schedule [88] [89] (see section 2.3). Although slowly cooling logarithmic schedules are generally considered more robust, we have adopted a geometric schedule. The principle reason for this choice was simply, the complexity of the algorithms makes only a relatively small number of iterations possible within a reasonable time-span, hence a fast annealing schedule is necessary. The schedule is given by,

$$T_t \;=\; (1 + \alpha_1)^{\alpha_2(1 - \frac{t}{N_t})} \tag{4.53}$$

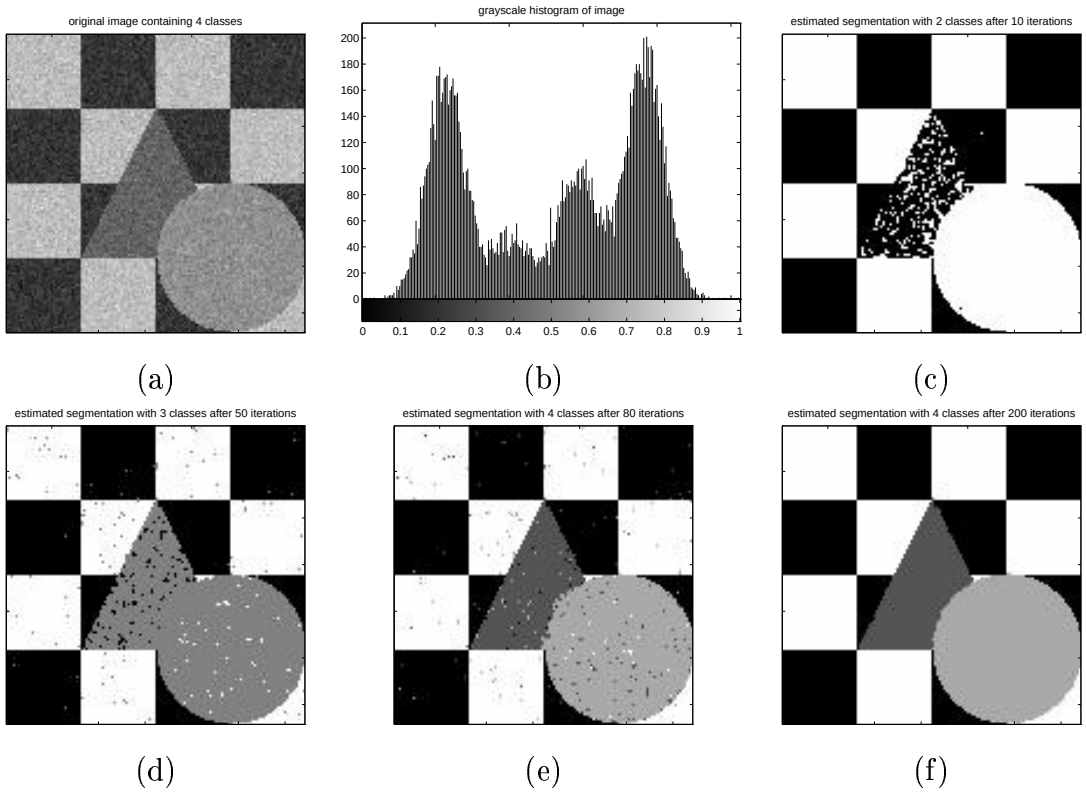


Figure 4.1: *200 iteration experiment on an image with four classes and beginning the algorithm with an arbitrary initial guess of two classes.*

where t is the iteration number, N_t is the total number of iterations, and α_1 and α_2 are constants (which were arbitrarily set to 0.1 and 10.0, respectively).

The isotropic segmentation algorithm described in the previous section, section 4.5, has been applied to various computer synthesised mosaics. For the purposes of these experiments, β the double pixel clique coefficient, which prescribes the interaction strength between pixels within the image, was set *a priori* to a value of 1.5. In proposing a reversible jump, the Kolmogorov-Smirnov distances were calculated using 9×9 windows to generate 40 bin histograms of the pixel gray-scale distribution functions.

Figure 4.1 shows the convergence of a 200 iteration run on a 4 class image. The segmentation is good, but the gray-level histogram demonstrates that the classes would be relatively well separated in the feature space discussed in chapter 3 and thus previous algorithms would be capable of achieving good block based segmentations. However, after few iterations, the number of classes has been correctly estimated.

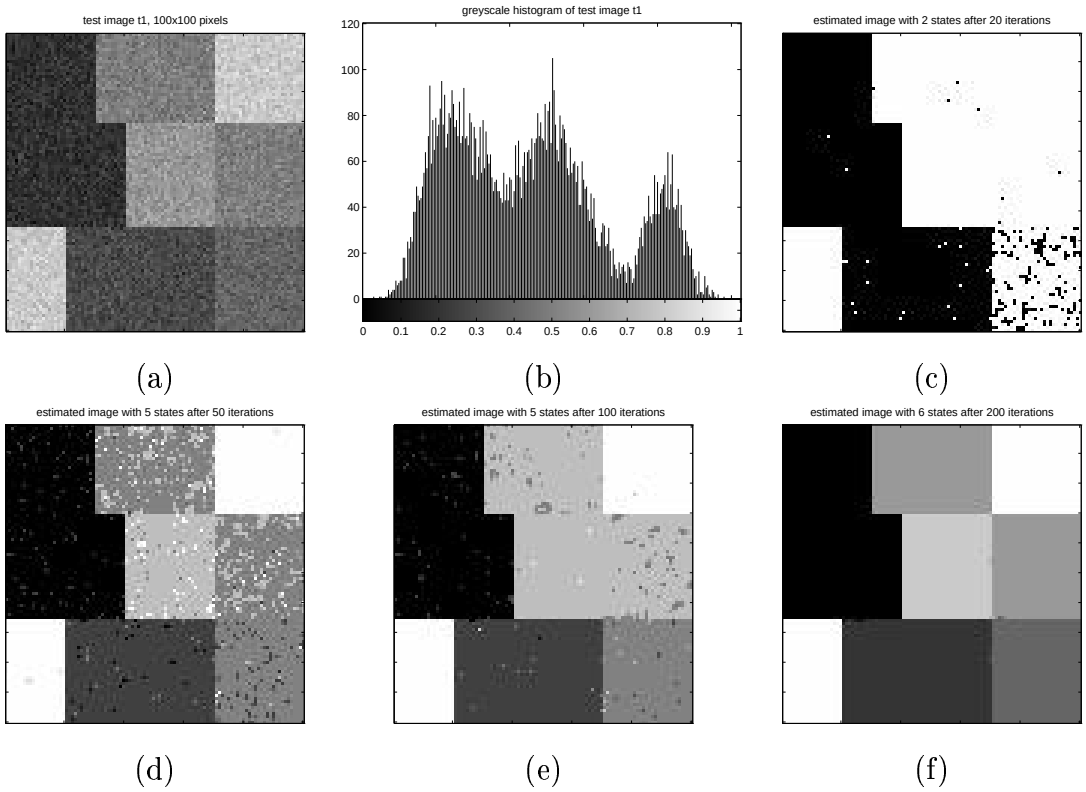


Figure 4.2: *200 iteration experiment on an image with six classes and beginning the algorithm with an arbitrary initial guess of two classes.*

Figure 4.2 shows the convergence of a 200 iteration run on a less clearly defined 6 class image. The segmentation is a good representation of the image after so few iterations and the number of classes has been correctly diagnosed. The results of a 500 iteration run are shown in figure 4.3 (see the end of this chapter for this and further results). The original gray-scale densities of the five classes are far closer than in the previous example (as shown by the gray-scale histogram) but the algorithm has still converged to a good estimate of the underlying image.

A grey-scale image of a house is segmented in Figure 4.4. The algorithm was repeated from various starting conditions and the results reached remained consistent. It could be argued that a better segmentation was arrived at in the intermediate steps but the algorithm then fits to a statistically if not visually better model in the final iterations. The possibility of using an information criterion to arrive at a more useful segmentation could be considered.

The hierarchical GMRF segmentation algorithm described in section 4.5 has been tested

parameters	class	μ	σ	θ_1	θ_2	θ_3	θ_4
actual	1	0.50	0.10	-0.10	0.20	0.20	0.20
	2	0.50	0.10	0.34	0.32	-0.21	-0.25
	3	0.50	0.10	0.20	0.20	-0.20	0.20
	4	0.50	0.10	0.20	0.20	0.20	-0.20
estimated	1	0.506706	0.094079	-0.119017	0.217379	0.209426	0.200397
	2	0.499145	0.10062	0.186007	0.18488	-0.182542	-0.116756
	3	0.501094	0.096016	0.197196	0.195557	-0.200425	0.18765
	4	0.511185	0.10211	0.173685	0.216785	0.194777	-0.197734

Table 4.1: *Actual and estimated parameters for the GMRF segmentation.*

on various computer generated mosaics of GMRF's. The neighbourhood size of each GMRF required four θ coefficients and so was of type $n = 2$ [54]. An example of such a test is shown in figures 4.5. Again the β parameter has been set *a priori* to 1.5. The algorithm was run for 600 iterations from a starting temperature of 2 to a finishing temperature of 0.1. The temperature schedule for the experiment was simply linear:

$$T_t = \frac{(N_t - t)}{N_t} * (T_{max} - T_{min}) + T_{min} \tag{4.54}$$

where t is the iteration number, N_t is the total number of iterations, and T_{min} and T_{max} are the minimum and maximum temperatures of the schedule. When estimating a likely reallocation of the label field during reversible jump proposal generation, windows of 16×16 pixels were used. The estimated parameters are compared to those used when generating the original image in Table 4.1. These estimates appear reasonable, given the size of the data available and some inaccuracy is inevitable due to the use of the pseudo-likelihood approximation.

In all these simulations, it is interesting to observe how at high temperatures the modelling is limited to the interaction between the field and the observed data. This is in direct parallel to the action of an external magnetic field on the Ising model for a ferromagnetic material, as described by Baxter[7]. In this case, long range correlations are expected to occur only as the system is cooled to around its critical temperature; field configurations are driven mainly by the external field at high temperature. An identical effect is evident when observing Figures 4.3 and 4.5: only as the temperature cools do the random fluctuations within the model begin to recede. A direct consequence of this

phenomena is the initial fitting at high temperatures of the model to the observed data's gray level histogram. It appears that at a particular temperature, the MRF prior is weak, and only when the temperature fall below that of criticality for the system, do inter-site interactions exert an affect. Hence, for example, the initial fitting of a pair of Gaussians to the image in figure4.3(a) and a single Gaussian distribution to the data in Figure 4.2(c). This, to date unreported process, makes the results of traditional unsupervised annealing algorithms of section 2.5 (i.e. where the number of classes is known but the associated parameters are not) appear at best, somewhat fortuitous. Without the introduction of reversible jumps, it is apparent that these segmentation experiments could, with reasonable probability, converge to a number of classes far fewer than the MAP estimate.

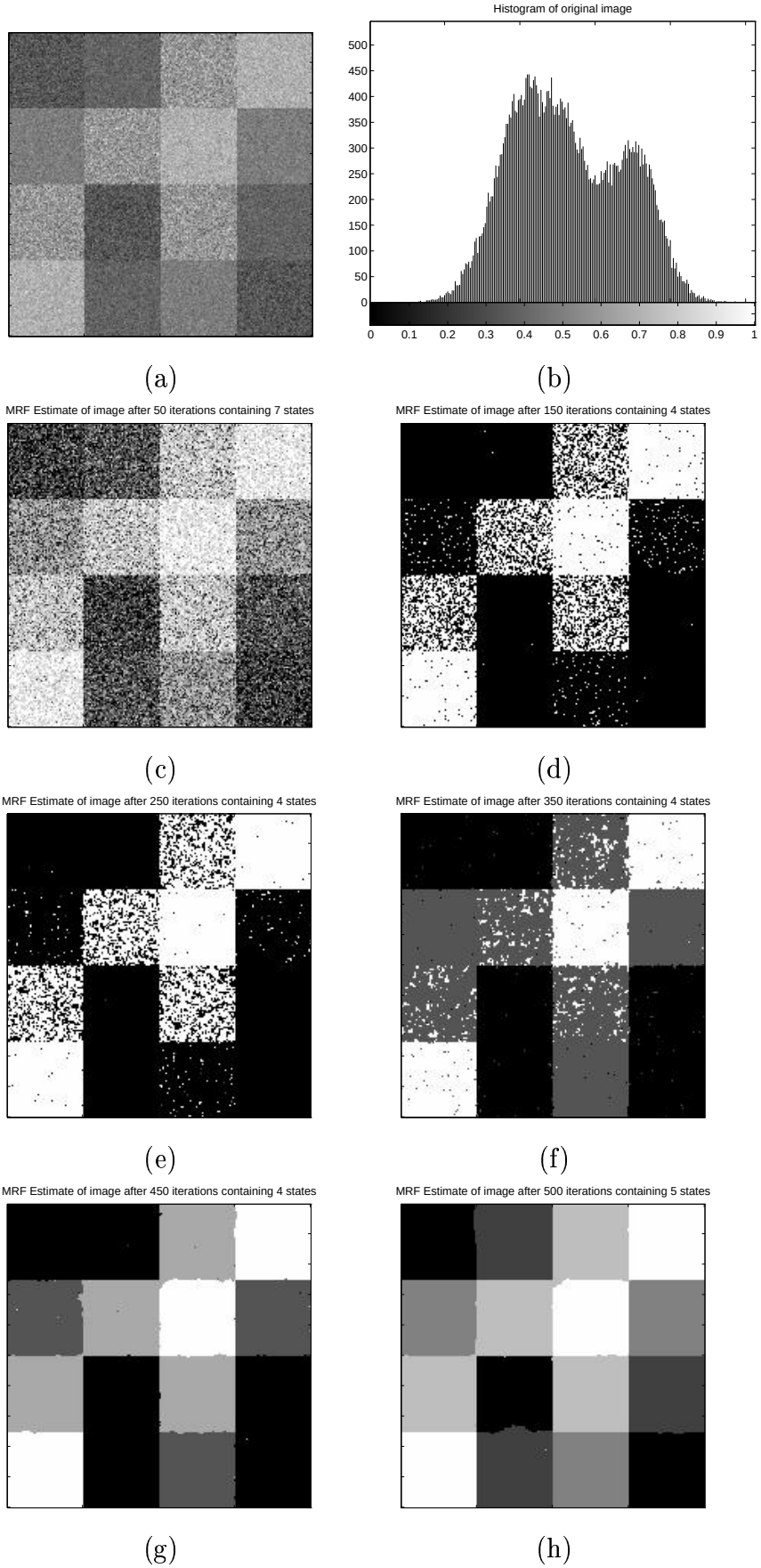


Figure 4.3: *500 iteration experiment on an image with five classes and beginning the algorithm with an arbitrary initial guess of six classes*

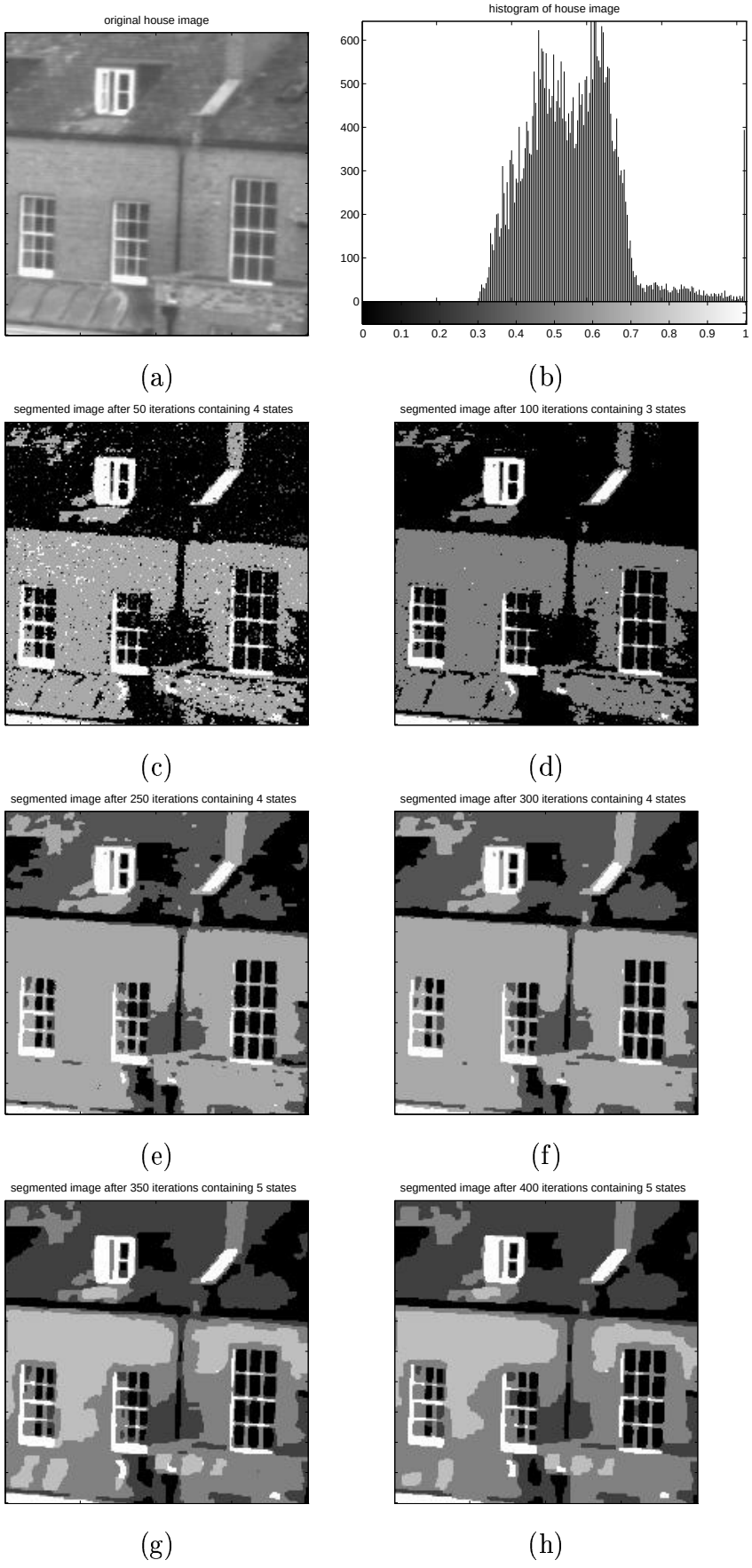


Figure 4.4: *400 iteration experiment on a gray-scale image of a house. The algorithm begins with an arbitrary initial guess of six classes and consistently converges to five.*

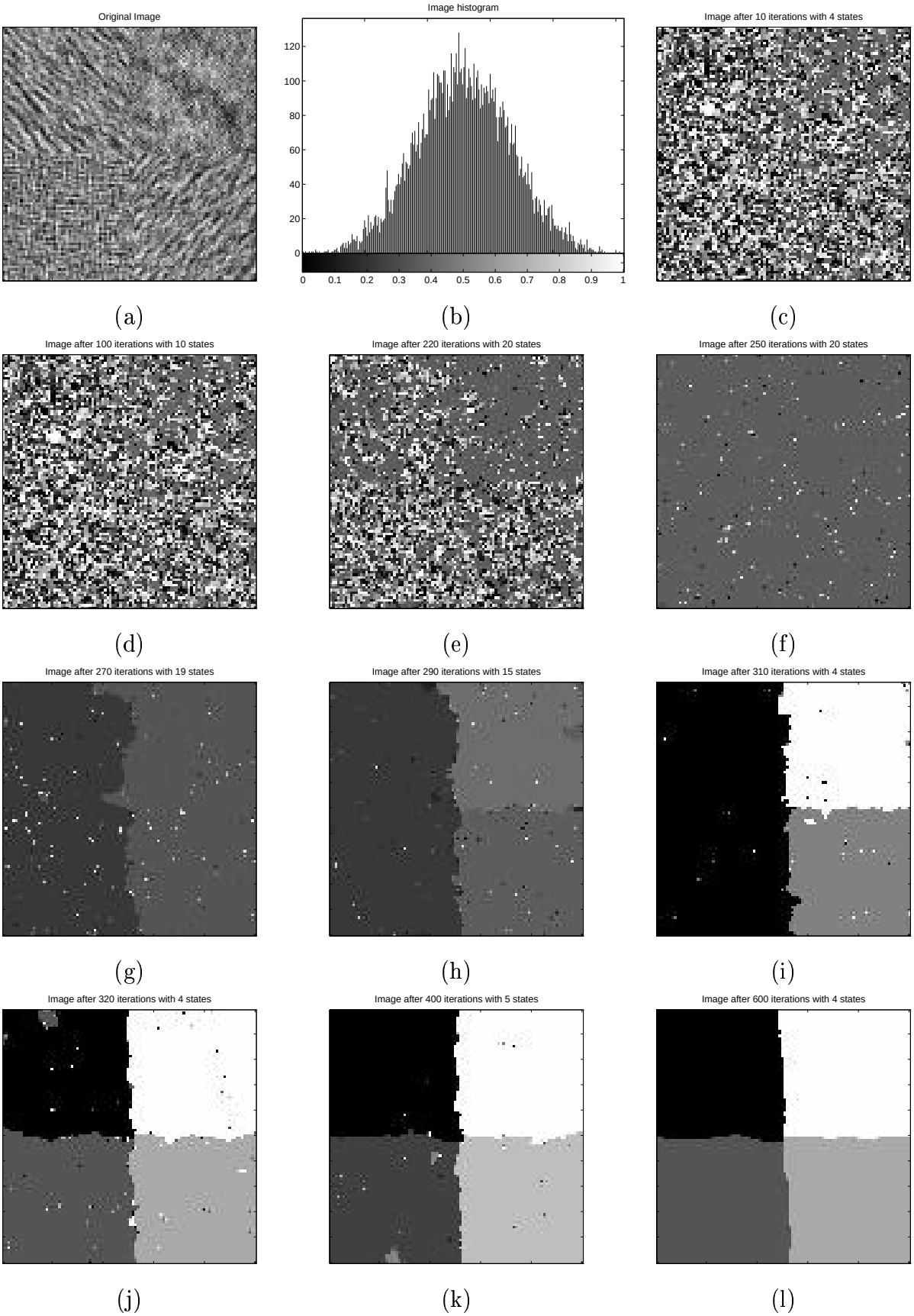


Figure 4.5: *600 iteration experiment on a four state synthetic GMRF image. The algorithm begins with an arbitrary initial guess of three classes and converges to four.*

4.7 Conclusion

In this chapter, unsupervised segmentation algorithms for both noisy and textured images have been presented. The models used, specifically hierarchical MRF's with Isotropic and GMRF region models, have been defined. Simulated annealing was used to optimise the model parameters, number of classes and pixel labels, thus requiring a MCMC sampling algorithm capable of drawing these parameters from the model's posterior distribution. Gibbs sampling, the Metropolis-Hastings algorithm and Reversible Jump MCMC were all applied to enable rapid and complete exploration of the combined model space.

Experimental results have been presented in which synthesised images have been rapidly segmented to give accurate estimates of the original underlying images. A real world image was also successfully segmented. The results demonstrated unexpected convergence features which justify the use of reversible jump MCMC when utilising simulated annealing to achieve image segmentation, even when the number of regions is known *a priori*.

However, a particular drawback of these algorithms is their computational intensity. The use of composition sampling to sample the order of the hierarchical GMRF model somewhat alleviates this problem, but to segment real world textures using such a model is beyond the scope of these algorithms. This is due to the large neighbourhoods required to simulate textures (such as Brodatz [16] textures) using GMRF models. The following chapter will address this problem, together with that of improving the convergence and the robustness of the optimisation process.

Partial Decoupling 5

5.1 Introduction

A major drawback to the two algorithms presented in the previous chapter is their slow speed of convergence. This can be attributed to the random nature in which new model parameters are proposed when attempting to create a new state and results in progressively lower rates of acceptance for increasingly more complex models. Despite the use of marginal distributions to minimise the number of new model parameters being proposed in each reversible jump, this problem appears to be ineluctable using the current algorithms. To overcome this, a new method of generating proposals[1] is described which originates from posing the question, 'can the data be used in the proposal of new states ?' To answer this question is complex. Firstly, a viable mechanism for generating new model parameters based on the data must be found and secondly, the resulting algorithm must maintain detailed balance to ensure that samples continue to be drawn from the target distribution.

To elaborate, consider the process of splitting one class into two: two new sets of parameters need to be generated. In the previous chapter, the new parameters were proposed randomly while preserving the model's central moments. The proposition here is to generate the new parameters based upon estimates from the data. To achieve this the data must first be clustered into groups which when combined comprise the region labelled by the class which the algorithm proposes to split. These groups or clusters of data may then be used to generate deterministically separate sets of model parameters, (possibly using MAP or maximum likelihood estimates) a selection of which, or a combination of which may be used to propose the new model parameters. It would appear profitable to partition the

[1] The work included in this chapter was presented at the 'International Conference on Acoustics, Speech and Signal Processing 1998', Seattle and was published in the conference proceedings[6].

data using some form of similarity measure, thus having the effect of clustering like data together and so guaranteeing the homogeneity of clusters while concurrently partitioning unlike data elements into separate clusters.

This process of parameter proposal comprises the fundamental difference between the algorithms presented here as opposed to those described in the previous chapter. The resulting algorithms achieve a fusion between low-level deterministic methods and those of higher level model based techniques, which has to date been seldom examined in literature.

To facilitate the partitioning of data into clusters, the Gibbs Sampler, used in the previous chapter to update the hidden states of the Markov Random Field, is replaced by a Partial-Decoupling algorithm[50], a derivative of the Swendsen-Wang algorithm[90]. The Swendsen-Wang algorithm was developed to give better mixing when sampling from a dynamic system at its critical temperature. The Partial-Decoupling algorithm is an improved version of Swendsen-Wang, designed to sample from a system with strong internal interactions which is experiencing a strong external field (as found when sampling from an MRF posterior distribution).

The remainder of this chapter comprises a further seven sections: section 5.2 is a review of the Swendsen-Wang and Partial Decoupling algorithms; section 5.4 describes an unsupervised segmentation algorithm for the Isotropic MRF, based upon the Partial Decoupling Algorithm; section 5.5 describes a similar algorithm for the Gaussian MRF; section 5.6 describes how line processes may be incorporated into the model; results are presented in section 5.7 and the chapter is concluded in section 5.8.

5.2 The Swendsen-Wang Algorithm

Geman & Geman[39] proved that convergence to a target distribution by Gibbs sampling was guaranteed in infinite time. Convergence to a MAP estimate using the simulated annealing algorithm was also shown to be of probability one in infinite time. These results, though useful in the theoretical sense have little practical importance in many multivariate optimisation problems. Far more important is the speed of mixing around the critical temperature of the dynamic system in question. It is well known that Gibbs sampling is extremely slow mixing when long range correlations occur at the critical temperature.

This is due to the single site update nature of the Gibbs Sampler.

To address this problem, various auxiliary variable schemes have been suggested [32] [12] [72] to facilitate the construction of Markov chains that are faster mixing. In such a scheme the vector of sampled variables $\mathbf{x}$ is augmented by a vector of auxiliary variables $\mathbf{u}$ such that the joint distribution may be factored $\pi(\mathbf{x}, \mathbf{u}) = \pi(\mathbf{x})\pi(\mathbf{u} \mid \mathbf{x})$, thus preserving the target distribution in the marginal $\pi(\mathbf{x})$. A Markov chain may then be constructed by alternately updating $\mathbf{x}$ and $\mathbf{u}$. Samples drawn from the joint distribution will also satisfy the marginal.

It is important to note that when applying auxiliary variable sampling schemes in the context of a simulated annealing algorithm, the objective is to find $\hat{\mathbf{x}} = \max_{\mathbf{x}\in\mathbf{X}} \pi(\mathbf{x})$, not $\hat{\mathbf{x}}, \hat{\mathbf{u}} = \max_{\mathbf{x}\in\mathbf{X},\mathbf{u}\in\mathbf{U}} \pi(\mathbf{x}, \mathbf{u})$. Thus the auxiliary variables are introduced to facilitate faster exploration at each temperature of the annealing algorithm, not to be annealed themselves.

The Swendsen-Wang algorithm[90] may be viewed as an auxiliary variable sampling process [50]. The algorithm was introduced to reduce relaxation times when simulating phase transitions in a Potts model. The model may be expressed in terms of a Gibbs distribution for a set of pixels defined on a lattice $\mathbf{\Omega}$. The distribution's parameters α and β may or may not be location dependent and so are indexed by site:

$$\pi(\mathbf{x}) \;\propto\; \exp\left\{\sum_{i\in\mathbf{\Omega}} \alpha_i(x_i) \;+\; \sum_{\{i,j\}\in\mathbf{\Omega}} \beta_{ij}\mathrm{I}[x_i = x_j]\right\} \tag{5.1}$$

where $I[\cdot]$ is the indicator function, taking values from $\{0, 1\}$. The first term of the above equation constitutes the coupling of an external field to the system and the second term describes the interactions between neighbouring sites within the model. The Swendsen-Wang algorithm introduces auxiliary bond variables to decouple the effect of the external field from the internal interactions within the model.

To preserve the Gibbs distribution as the marginal $\pi(\mathbf{x})$, the bond variables u_{ij} are defined to be independently and uniformly distributed and conditioned on $\mathbf{x}$ such that,

$$p(u_{ij} \mid \mathbf{x}) \;\propto\; \exp\{-\beta_{ij}\mathrm{I}[x_i = x_j]\} \;\; \mathrm{I}\big[0 \le u_{ij} \le \exp\{\beta_{ij}\mathrm{I}[x_i = x_j]\}\big] \tag{5.2}$$

Hence, if $x_i = x_j$ then the probability the two sites are bonded, i.e. $u_{ij} > 1$ will equal $1 - \exp\{-\beta_{ij}\}$. Applying Bayes' theorem $\pi(\mathbf{x} \mid \mathbf{u}) \;\propto\; p(\mathbf{u} \mid \mathbf{x})\pi(\mathbf{x})$, gives the distribution for $\mathbf{x}$ conditioned on $\mathbf{u}$:

$$\pi(\mathbf{x} \mid \mathbf{u}) \;\propto\; \exp\left\{\sum_{i\in\mathbf{\Omega}} \alpha_i(x_i)\right\} \prod_{i,j\in\mathbf{\Omega}} \mathrm{I}\big[0 \le u_{ij} \le \exp\{\beta_{ij}\mathrm{I}[x_i = x_j]\}\big] \tag{5.3}$$

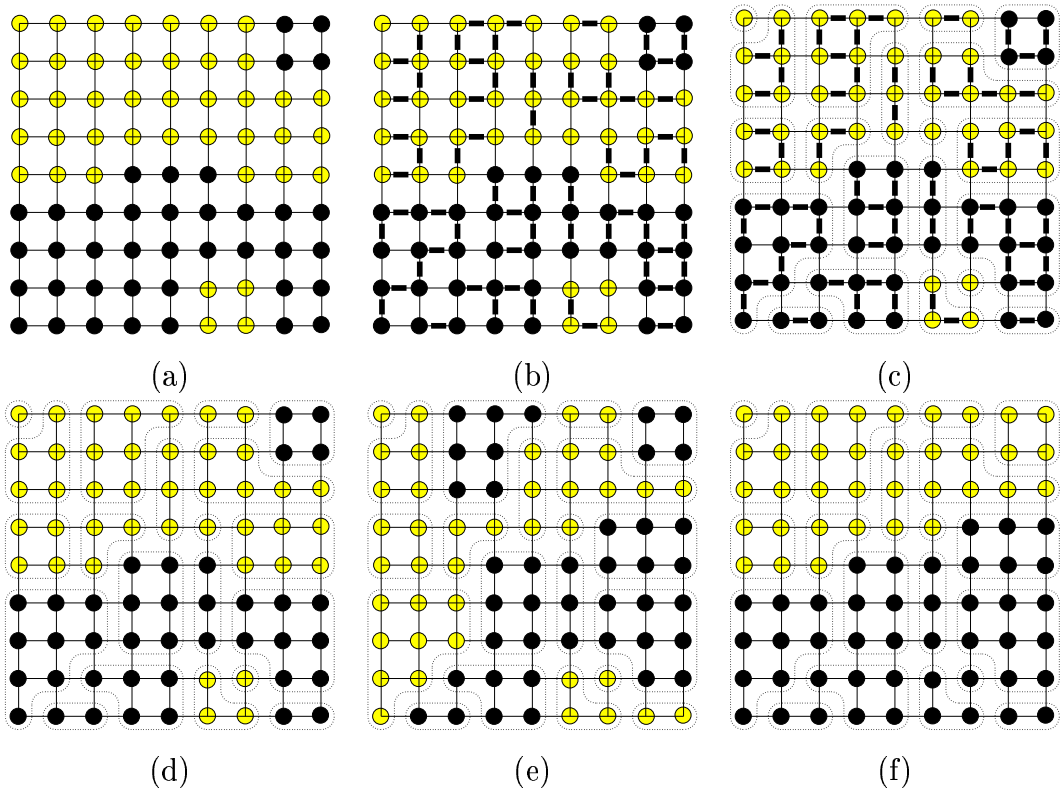


Figure 5.1: *Illustration of the operation of the Swendsen-Wang and Partial Decoupling algorithms: (a) current states, (b) bonds sampled given the states, (c) bonds and corresponding clusters, (d) current clusters, (e) states sampled given the bonds using the Swendsen-Wang algorithm, (f) states sampled given the bonds using the Partial Decoupling algorithm.*

When sampling from this distribution, to keep the product of indicator functions non-zero necessitates a sampling scheme that updates the pixel states in clusters formed by the bond variable configurations, otherwise the homogeneity of bonded pixels cannot be maintained. Hence, if the bond configuration designates that a set of pixels forms a cluster C, then the distribution for their states conditioned on the bonds is

$$p(x_i = k, i \in C \mid \mathbf{u}) \;\propto\; \exp\left\{ \sum_{i \in C} \alpha_i(k) \right\} \tag{5.4}$$

Thus, in summary, the Swendsen-Wang algorithm operates as shown in figure 5.1. Given the current configuration of states, figure 5.1(a), the bond variables are sampled from their conditional distributions of equation 5.2, giving configurations typical of that shown in figure 5.1(b). The bond variables define clusters, see figure 5.1(c), which are then

sampled independently as from equation 5.4, thus giving a recolouring of the as shown in figures 5.1(d) and (e).

Although more successful than single site update schemes, the Swendsen-Wang algorithm has been found to perform poorly in the presence of an external field when interactions are strong [72] [50] [49] [42]. The method of simulated tempering [72] [98] has been shown to be a large improvement for sampling an Ising model at critical temperature. However, this is not easily applied to solving multivariate optimisation problems via simulated annealing, since the basis of the algorithm, uniformly sampling the temperature parameter as an auxiliary variable makes the combination of the two algorithms intractable.

Higdon [49] proposed a derivative of Swendsen-Wang, the partial decoupling algorithm, to address the mixing deficiencies in the presence of an external field. The algorithm may be viewed as a generalisation of Swendsen-Wang with Gibbs sampling and pure Swendsen-Wang its two extremes. As its name suggests, the external field is only partially decoupled from the internal interactions by bond variables whose conditional distributions are given by

$$p(u_{ij} \mid \mathbf{x}) \;\propto\; \exp\{-\delta_{ij}\beta_{ij}\mathrm{I}[x_i = x_j]\}\;\; \mathrm{I}\big[0 \leq u_{ij} \leq \exp\{\delta_{ij}\beta_{ij}\mathrm{I}[x_i = x_j]\}\big] \tag{5.5}$$

The degree of decoupling is determined by the δ_{ij} terms which may or may not be site specific. To maintain the Gibbs distribution 5.1 as the target distribution, the application of Bayes' theorem results in a conditional distribution for $\mathbf{x}$:

$$\begin{aligned} \pi(\mathbf{x} \mid \mathbf{u}) \;\propto\; \exp\Bigg\{ \sum_{i \in \mathbf{\Omega}} \alpha_i(x_i) \;+\; \sum_{i,j \in \mathbf{\Omega}} (1 - \delta_{ij})\beta_{ij}\mathrm{I}[x_i = x_j] \Bigg\} \\ \times \prod_{i,j \in \mathbf{\Omega}} \mathrm{I}\big[0 \leq u_{ij} \leq \exp\{\delta_{ij}\beta_{ij}\mathrm{I}[x_i = x_j]\}\big] \end{aligned} \tag{5.6}$$

Comparing this to the equivalent distribution for the Swendsen-Wang algorithm (see equation 5.3) it is apparent that the distribution for the state of each cluster is now incompletely decoupled from its surrounding pixels. Alternatively, the probability for each cluster is conditional on the state of surrounding pixels. The degree of dependency is governed by the δ_{ij} parameters and in the limit, as $\delta_{ij} \to 1$, the algorithm approaches Gibbs sampling. Alternatively, as $\delta_{ij} \to 0$, the closer the algorithm to Swendsen-Wang. The corresponding conditional distribution for the colour or state allocated to a cluster C, is now given by

$$p(x_i = k, i \in C \mid \mathbf{u}) \;\propto\; \exp\Bigg\{ \sum_{i \in C} \alpha_i(k) \;+\; \sum_{i \in C, j \in \partial C} (1 - \delta_{ij})\beta_{ij}\mathrm{I}[x_j = k] \Bigg\} \tag{5.7}$$

where ∂C consists of the set of surrounding pixels which are nearest-neighbours of any pixels contained in the cluster C but are not themselves elements of C.

So to summarise, the partial decoupling algorithm operates as shown in figure 5.1. The operation is much the same as previously described for the Swendsen-Wang algorithm except bonds are sampled from conditionals of the type given by equation 5.5 and clusters are coloured, conditioned on neighbouring sites according to equation 5.7. Hence given the cluster configuration of figure 5.1(d), a likely recolouring using the partial decoupling algorithm is shown in figure 5.1(f).

The order in which cluster colours are sampled is an issue that needs to be addressed when utilising the Swendsen-Wang algorithm. The alternatives would appear to be either a form of raster scanning, similar to that used in the Gibbs sampler, or a scheme that updates clusters in an order based on some measure of cluster size.

The localisation of the δ_{ij} parameters allows an extra degree of versatility to be incorporated into the sampling scheme. To prevent clusters becoming detrimentally large, thus irrevocably slowing to the mixing process, the lattice can be broken into sub-lattices by setting rows and columns of δ_{ij} parameters to zero.

A further idea, useful when sampling from an MRF posterior distribution (and of paramount importance in this chapter) is the linking of the δ_{ij} parameters to the observed data. In a maximum *a posteriori* optimisation problem the interaction of the observed data with the model states corresponds to the action of the external field on the Ising or Potts models. Thus, if the observed data is denoted $\mathbf{y}$, then by setting $\delta_{ij} = \frac{1}{1+|y_i - y_j|}$, bonding is encouraged in regions where the data is homogeneous.

5.2.1 Extension to the general Potts Model

The partial decoupling algorithm is applicable to the general Potts model expressed in terms of a general potential function $V(x_i, x_j)$ (as opposed to the indicator function $\mathrm{I}[\cdot]$). The potential function takes values from $\{-1, 1\}$ to model interactions between neighbouring pixels. Hence, the conditional distribution for the auxiliary bond variables will be given by

$$p(u_{ij} \mid \mathbf{x}) \;\propto\; \exp\{-\delta_{ij}\beta_{ij}V_c(x_i, x_j)\}\;\; \mathrm{I}\big[0 \leq u_{ij} \leq \exp\{\delta_{ij}\beta_{ij}V_c(x_i, x_j)\}\big] \tag{5.8}$$

and the probability of two neighbouring sites being bonded, i.e. $u_{ij} \geq \exp\{-\delta_{ij}\beta_{ij}\}$, will be $1 - 2\exp\{-\delta_{ij}\beta_{ij}\}$. The conditional distribution for $\mathbf{x}$ given the bond configuration is

$$\begin{aligned} \pi(\mathbf{x} \mid \mathbf{u}) \;\propto\; \exp\Bigg\{ \sum_{i \in \mathbf{\Omega}} \alpha_i(x_i) \;+\; \sum_{i,j \in \mathbf{\Omega}} (1 - \delta_{ij})\beta_{ij} V(x_i, x_j) \Bigg\} \\ \times \prod_{i,j \in \mathbf{\Omega}} \mathrm{I}\big[0 \leq u_{ij} \leq \exp\{\delta_{ij}\beta_{ij} V(x_i, x_j)\}\big] \end{aligned} \tag{5.9}$$

Equation 5.7 may be similarly generalised.

5.3 Overview of Approach

The general optimisation process used in this chapter is simulated annealing, thus the process is iterative and consist of the following steps:

---

1. generate auxiliary variable hyper-parameters δ_{ij},
2. randomly initialise auxiliary bonds and hidden variables,
3. set $T = T_{MAX}$
4. sample the current auxiliary bond configuration given the hidden label configuration,
5. sample hidden label configuration given the current auxiliary bond configuration,
6. sample region likelihood model parameters,
7. sample the number of classes,
8. lower T and if $T > T_{MIN}$ goto step (4).

Each of these steps will be described in more detail, firstly for the hierarchical Isotropic MRF, in section 5.4 and then for the hierarchical GMRF model, section 5.5. The addition of a further line process, of the type reviewed earlier in section 2.2.2 is then discussed in section 5.6.

5.4 The Isotropic MRF

The Isotropic Markov Random Field image model used in this chapter is identical to that described for use in the previous chapter (see section 4.4.1), hence the definitions and notations of the previous chapter are maintained here.

5.4.1 Sampling the labels via the Partial Decoupling Algorithm

As outlined in the introduction to this chapter, when proposing to split a region into two a pair of new model parameter vectors must be estimated. Previously (see section 4.5.1) these have been randomly sampled, but here we propose to estimate new model parameters from the data. To achieve this the data must be grouped into sub-groups or clusters from which model parameters can then be estimated using ML or MAP procedures. If these clusters are generated according to some form of similarity measure so that like pixels form clusters, then the increased probability of homogeneity across each cluster will ensure the new parameter vectors are representative of these local areas of the image.

To illustrate this point, consider the case where a single class is incorrectly modelling what should constitute two different regions, then to propose a re-segmentation or splitting of the region, two different clusters or sub-groups of the region may be chosen from which two new sets parameters can be estimated. These new parameters may then be used as the basis for re-segmenting the complete region originally modelled by the single class.

To achieve data clustering or 'partitioning' when considering a Markov Random Field, an obvious candidate algorithm would be to sample labels using Swendsen-Wang(see section 5.2). The algorithm gives improved mixing when sampling from a Markov Random Field when its temperature is close to critical. To achieve this, auxiliary bond variables are incorporated into the model, which partition the image into local clusters, whose state or class may then be sampled independently. If a derivative of Swendsen-Wang is used, the partial decoupling algorithm, then pixels may be grouped into clusters or regions based upon a set of local parameters at each pixel. If these parameters are made indicative of a similarity measure on the observed data, then like pixels will be encouraged to cluster more readily, generating homogeneous regions. The partial decoupling algorithm was intended to give improved mixing over Swendsen-Wang, particularly when the system is composed

of strong internal interactions and is in the presence of a strong external field. When sampling pixel classes in an image segmentation problem, the external field is mirrored by likelihood terms linking the observed data to the underlying field and so by using the partial decoupling algorithm, rates of convergence may generally be expected to be higher than those found when using an equivalent Gibbs sampling algorithm (e.g. the algorithms of the previous chapter).

Recalling the partial decoupling algorithm's conditional distributions for the Potts model 5.8 and 5.9: these may be re-written to allow the algorithm to sample the labels of the Isotropic Markov Random Field model (as defined in section 4.4.1). The resulting conditional distributions are

$$p(u_{ij} \mid \mathbf{x}, \boldsymbol{\delta}, \beta) \propto \exp\{-\delta_{ij}\beta V(x_i, x_j)\} \ \mathrm{I}[0 \leq u_{ij} \leq \exp\{\delta_{ij}\beta V(x_i, x_j)\}] \tag{5.10}$$

$$p(\{x_i = c, i \in \boldsymbol{\mathcal{C}}\} \mid \mathbf{u}, \boldsymbol{\delta}, \beta, \mu_c, \sigma_c, \mathbf{y}) \propto \exp\left\{ \sum_{i \in \boldsymbol{\mathcal{C}}} \Big[-\frac{1}{2}\Big(\frac{y_i - \mu_c}{\sigma_k}\Big)^2 + \sum_{j \in \partial\boldsymbol{\mathcal{C}}} (1 - \delta_{ij})\beta V(x_i, x_j) \Big] \right\} \tag{5.11}$$

where: $\partial\boldsymbol{\mathcal{C}}$ is the ring of pixels around the cluster $\boldsymbol{\mathcal{C}}$, each of which is a neighbour of at least one pixel in C; $\boldsymbol{\delta}$ is the matrix of bond variable hyper-parameters δ_{ij}, specifying the prior probability of the formation of the bond u_{ij} linking pixels i and j; I[ ] is the indicator function. To quantify this relationship, the probability that two sites are said to be bonded, i.e. $u_{ij} \geq \exp\{-\delta_{ij}\beta\}$, is $1 - 2\exp\{-\delta_{ij}\beta\}$. Hence, the greater the value of δ_{ij}, the greater the probability of the formation of a bond.

The choice of a criteria for generating a matrix of values for $\boldsymbol{\delta}$ is arbitrary. The Kolomogorov-Smirnov (KS) distance has been used as a measure of difference between the distributions of gray scale values in other segmentation algorithms[37][61]. Geman *et al*[37] used the distance to give a measure of similarity between distributions in overlapping blocks. These distances were then applied to form the basis of their unsupervised segmentation algorithm. Here the distance is evaluated in a similar fashion, but it is used as a hyper-parameter for an auxiliary bond variable and thus gives the prior probability of the two pixels at the centre of each block being bonded. To maintain a consistency of bond proliferation throughout the simulated annealing cooling process when applying the algorithm to different images, it is desirable to normalise the distribution of the bond hyper-parameters for mean and standard deviation. Hence, if $d_{ij}^{(ks)}$ is the KS distance between

blocks centred at pixels i and j, then the bond hyper-parameter mean $\hat{\mu}_{ks} = \left\langle \frac{1}{1+d_{ij}^{(ks)}} \right\rangle$, over the image lattice and standard deviation $\hat{\sigma}_{ks} = \left\langle \left(\frac{1}{1+d_{ij}^{(ks)}} - \mu_{ks} \right)^2 \right\rangle$ must be found. This allows the normalisation of the distribution of δ_{ij} parameters via the equation,

$$\delta_{ij} \;=\; \left(\frac{1}{1+d_{ij}^{(ks)}} - \hat{\mu}_{ks} \right) \frac{\sigma_\delta^{(KS)}}{\hat{\sigma}_{ks}} \;+\; \mu_\delta^{(KS)} \tag{5.12}$$

where $\mu_\delta^{(KS)}$ and $\sigma_\delta^{(KS)}$ are pre-specified parameters defining the desired distribution of the δ_{ij} parameters.

5.4.2 Parameter Estimation

Model parameters are sampled from their posterior distributions in a similar fashion to the algorithms presented in the previous chapter. Simple Metropolis sampling is used to update parameter estimates from their posterior distribution. Priors are identical to those used in the previous chapter being non-informative: i.e. uniform for the location parameters and Jeffrey's for the scale parameters. The resulting posterior distribution is given by,

$$\begin{aligned} p(\mu_c, \sigma_c \mid \mathbf{y}) \quad &\propto \quad \prod_{\boldsymbol{c} \in \boldsymbol{\Delta}_c} \prod_{i \in \boldsymbol{c}} p(y_i \mid \mu_c, \sigma_c)\; p_r(\mu_c)\; p_r(\sigma_c) \\ &\propto \quad \frac{1}{\sigma_c\; (2\pi\sigma_c^2)^{N_c}} \exp\left\{ -\frac{1}{2} \sum_{\boldsymbol{c} \in \boldsymbol{\Delta}_c} \sum_{i \in \boldsymbol{c}} \left(\frac{y_i - \mu_c}{\sigma_c} \right)^2 \right\} \end{aligned} \tag{5.13}$$

where $\boldsymbol{\Delta}_c$ is the set of clusters allocated to class c, N_c is the number of pixels allocated to class c and $p_r(\cdot)$ denotes a prior distribution.

5.4.3 Reversible Jumps for the Isotropic MRF

To propose a reversible jump move which will split one class c, into two requires the generation of two new sets of model parameters. The region labelled by this class comprises pixels which have previously been grouped (by the partial decoupling algorithm) into clusters. Two of the largest N_c of these clusters are randomly selected to generate two sets of maximum likelihood estimates of model parameters. These parameters are then used to propose a new segmentation of all pixels allocated to the existing, single class into regions labelled by the two new classes.

New model parameters are proposed by calculating maximum likelihood estimates based on image data corresponding to a randomly selected pair of clusters $\mathcal{C}_{c1}$ and $\mathcal{C}_{c2}$, from the original class. The estimated parameter vectors are denoted $\hat{\boldsymbol{\phi}}_{c1} = [\hat{\mu}_{c1}, \hat{\sigma}_{c1}]$ and $\hat{\boldsymbol{\phi}}_{c1} = [\hat{\mu}_{c2}, \hat{\sigma}_{c2}]$. However, to allow the reverse transition to be possible, thus satisfying the condition for detailed balance, a small random perturbation is added to the estimate, hence the new model parameter vectors, $\boldsymbol{\phi}_{c1} = [\mu_{c1}, \sigma_{c1}]$ and $\boldsymbol{\phi}_{c2} = [\mu_{c2}, \sigma_{c2}]$ are proposed according to

$$\begin{aligned}
\hat{\mu}_{c1} &= \tfrac{1}{N_{c1}} \textstyle\sum_{i \in \mathcal{C}_{c1}} y_i & \qquad \hat{\mu}_{c2} &= \tfrac{1}{N_{c2}} \textstyle\sum_{i \in \mathcal{C}_{c2}} y_i \\
\mu_{c1} &= \hat{\mu}_{c1} + e_{m1} & \qquad \mu_{c2} &= \hat{\mu}_{c2} + e_{m2} \\
\sigma_{c1} &= \sqrt{\tfrac{1}{N_{c1}} \textstyle\sum_{i \in \mathcal{C}_{c1}} (y_i - \hat{\mu}_{c1})^2} + e_{s1} & \qquad \sigma_{c2} &= \sqrt{\tfrac{1}{N_{c2}} \textstyle\sum_{i \in \mathcal{C}_{c2}} (y_i - \hat{\mu}_{c2})^2} + e_{s2}
\end{aligned} \tag{5.14}$$

where, N_{c1} and N_{c2} are the sizes of the two clusters, and e_{m1}, e_{m2}, e_{s1} and e_{s2} are the perturbation random variables, drawn from zero mean i.i.d. Gaussian and shifted Gamma distributions to ensure positivity in the variance parameter proposals. The proposal probability for the new parameters is

$$\begin{aligned}
q(\boldsymbol{\phi}_{c1}, \boldsymbol{\phi}_{c2} \mid \mathbf{u}, \mathbf{y}) = \frac{1}{N_c(N_c - 1)} \sum_{i=1}^{N_c} \sum_{j=1:j\neq i}^{N_c} & p(e_{m1} = \mu_{c1} - \hat{\mu}_i)\; p(e_{m2} = \mu_{c2} - \hat{\mu}_j) \\
& \times p(e_{s1} = \sigma_{c1} - \hat{\sigma}_i)\; p(e_{s2} = \sigma_{c2} - \hat{\sigma}_j)
\end{aligned} \tag{5.15}$$

thus ML estimates of parameter vectors must be calculated for all N_c clusters considered.

The reverse of this move, the combining of two classes into one requires a single set of new parameters. These are proposed to preserve first and second order moments but with the addition of a small perturbation random variable, thus ensuring the reversibility condition is satisfied:

$$\hat{\mu}_c = \frac{N_{c1}\mu_{c1} + N_{c2}\mu_{c2}}{N_{c1} + N_{c2}} \tag{5.16}$$

$$\mu_c = \hat{\mu}_c + a_m \tag{5.17}$$

$$\sigma_c = \sqrt{\frac{N_{c1}(\mu_{c1}^2 + \sigma_{c1}^2) + N_{c2}(\mu_{c2}^2 + \sigma_{c2}^2)}{N_{c1} + N_{c2}} - \hat{\mu}_c^2} + a_s \tag{5.18}$$

where a_m and a_s are independent Gaussian random variables. Hence, the probability of proposing a combined parameter vector is simply $q(\boldsymbol{\phi}_c \mid \boldsymbol{\phi}_{c1}, \boldsymbol{\phi}_{c2}, \mathbf{x}) = p(a_m)\; p(a_s)$.

The above equations completely describe the bijection between the extended parameter vectors: $[\mu_c, \sigma_c, e_{m1}, e_{m2}, e_{s1}, e_{s2}] \leftrightarrow [\mu_{c1}, \mu_{c2}, \sigma_{c1}, \sigma_{c2}, a_s, a_m]$. The Jacobian determinant of this transformation, required in the calculation of the reversible jump acceptance ratio is simply 1.

The re-segmentation of all clusters allocated to class c, the class being split, may now be carried out by using the new model parameters. The methodology used is simple and is somewhat intuitive: clusters are re-allocated using the partial decoupling algorithm's cluster colouring equation 5.11. However, a state of ignorance exists when considering the conditioning of this equation on each cluster's neighbourhood configuration. To overcome this, all pixels of the split state are first assumed to belong to another, different and invalid state. Next, clusters are allocated to one of the new pair of states in order of size, the largest first, by Gibbs sampling from their relative conditional probabilities. Hence, the algorithm colours the largest clusters relatively independently but as the size of the clusters decrease and the gaps are filled between these large clusters, the clusters' colouring distributions become conditioned to an ever greater extent on their local neighbourhoods. The probability of proposing such a re-segmentation of the label field $\mathbf{x}$ to $\mathbf{x}^+$ is given by,

$$q(\mathbf{x}, \mathbf{x}^+) \;=\; \prod_{\boldsymbol{\mathcal{C}} \in \boldsymbol{\Delta}_c} \frac{p\big(\{x_i, i \in \boldsymbol{\mathcal{C}}\} \mid \mathbf{u}, \boldsymbol{\delta}, \beta, \boldsymbol{\phi}_{x_i}, \mathbf{y}\big)}{p\big(\{x_i = c1, i \in \boldsymbol{\mathcal{C}}\} \mid \mathbf{u}, \boldsymbol{\delta}, \beta, \boldsymbol{\phi}_{c1}, \mathbf{y}\big) \;+\; p\big(\{x_i = c2, i \in \boldsymbol{\mathcal{C}}\} \mid \mathbf{u}, \boldsymbol{\delta}, \beta, \boldsymbol{\phi}_{c2}, \mathbf{y}\big)} \tag{5.19}$$

where $\boldsymbol{\Delta}_c$ is the set of all clusters allocated to class c before the proposed split, and the distributions $p(x_i = k, i \in C \mid \mathbf{u}, \boldsymbol{\delta}, \beta, \boldsymbol{\phi}_k)$ are the cluster colouring partial decoupling conditional distributions as given by equation 5.11.

When considering the opposite move, the combining of two states into one, the reverse proposal probability (identical to that if proposing to split the clusters comprising the new merged state into the two original states) must be calculable. Because the order in which the clusters are re-allocated is deterministic, the back-calculation of the reverse allocation probability is possible via the above equation.

The resulting acceptance ratio for splitting the region labelled by class c into regions labelled by $c1$ and $c2$ is given by

$$\min\left[1,\; \frac{\pi(\mathbf{x}^+, \boldsymbol{\psi}^+ \mid \mathbf{y})}{\pi(\mathbf{x}, \boldsymbol{\psi} \mid \mathbf{y})} \; \frac{q(\text{merge})}{q(\text{split})} \; \frac{q(c1, c2)}{q(c)} \; \frac{q(\boldsymbol{\phi}_c \mid \boldsymbol{\phi}_{c1}, \boldsymbol{\phi}_{c2}, \mathbf{x})}{q(\boldsymbol{\phi}_{c1}, \boldsymbol{\phi}_{c2} \mid \mathbf{u}, \mathbf{y})} \; \frac{1}{q(\mathbf{x}, \mathbf{x}^+)}\right] \tag{5.20}$$

where the first term comprises the ratio of the posterior probabilities after and before the proposed split and the other terms are the proposal probabilities, as defined in the above

equations with the exception of $q(c)$ and $q(c1, c2)$: these are simply the probabilities of selecting the class or classes to be split or combined. As described, the Jacobian determinant corresponding to the transformation between extended parameter vectors is unity and hence is omitted from this expression. Thus, in summary, the split move of the reversible jump for the hierarchical Isotropic MRF model is given by the following steps:

---

1. randomly select with probability $\frac{1}{k}$ a state from $\mathbf{\Lambda}^{(k)}$ to split,
2. randomly select two of the largest N_c clusters $\boldsymbol{\mathcal{C}}_{c1}$ and $\boldsymbol{\mathcal{C}}_{c2}$, with probability $q(c1, c2)$,
3. estimate using ML or MAP criteria parameter vectors $\hat{\boldsymbol{\phi}}_{c1}$ and $\hat{\boldsymbol{\phi}}_{c2}$ from the observed data pertaining to clusters $\boldsymbol{\mathcal{C}}_{c1}$ and $\boldsymbol{\mathcal{C}}_{c2}$. Also estimate parameter vectors for the remaining clusters of the N_c under consideration for use in the calculation of the transition probability,
4. generate the two proposed parameter vectors by drawing the random vector $\mathbf{e}$ from the respective proposal distributions and substituting, together with the old parameter vector estimates $\hat{\boldsymbol{\phi}}_{c1}$ and $\hat{\boldsymbol{\phi}}_{c2}$ into the mapping functions contained in equation 5.14 and calculate $q(\boldsymbol{\phi}_{c1}, \boldsymbol{\phi}_{c2} \mid \mathbf{u}, \mathbf{y})$ according to equation 5.15,
5. calculate the probability of generating parameter vector $\boldsymbol{\phi}_c$ as if generating the reverse or merging proposal according to equations 5.16 to 5.18. This will correspond to the probability of generating the random vector $[a_m, a_s, a_t]$ as calculated by substituting the parameter values obtained in the previous step into equations 5.16 to 5.18,
6. propose a new segmentation of clusters previously allocated to state c into c_1 and c_2 using the iterative procedure described above, with probability $q(re - segmentation)$,
7. calculate the overall acceptance ratio according to equation 5.20 and accept or reject the proposal based upon this value.

The merge move follows the above steps but implementing the opposite moves, with reverse transition probabilities calculated as if following the above steps. Hence, the acceptance ratio for the combine move is simply the inverse of that in equation 5.20.

5.5 Gaussian MRF

The Gaussian Markov Random Field image model is identical to that described in the previous chapter (see section 4.4.2) hence the definitions and notations of the previous chapter are maintained throughout this section.

5.5.1 Partial Decoupling

The Gaussian MRF conditional distribution for the partial decoupling algorithm's auxiliary variables is identical to that of the more simple Isotropic MRF (see equation 5.10). However, the cluster colouring conditional distribution is far more difficult to obtain. The distribution may be factored

$$\begin{aligned} p\big(\{x_i = c, i \in \boldsymbol{\mathcal{C}}\} \mid \mathbf{u}, \boldsymbol{\delta}, \beta, \mu_k, \sigma_k, \mathbf{y}\big) \;\propto\; & p\big(\{y_i, i \in \boldsymbol{\mathcal{C}}\} \mid \{y_j, j \in \partial\boldsymbol{\mathcal{C}}\}, \mu_k, \sigma_k\big) \\ & \times\; p\big(\{x_i = k, i \in \boldsymbol{\mathcal{C}}\} \mid \mathbf{u}, \boldsymbol{\delta}, \beta\big) \end{aligned} \tag{5.21}$$

where the first term corresponds to the fit of the observed gray scale values of the pixels constituting the cluster and its neighbours to the model selected by state c. The second term models the interactions in the underlying state process between each cluster and its neighbours. This second term may be treated identically to the Isotropic case, i.e.

$$p\big(\{x_i = c, i \in \boldsymbol{\mathcal{C}}\} \mid \mathbf{u}, \boldsymbol{\delta}, \beta\big) \;\propto\; \exp\Big\{ \sum_{i \in \boldsymbol{\mathcal{C}}} \sum_{j \in \partial\boldsymbol{\mathcal{C}}} (1 - \delta_{ij}) \beta V(x_i, x_j) \Big\} \tag{5.22}$$

It should be noted that the local texture model is fitted to each cluster and its neighbouring pixels, rather than to just those pixels within the cluster. This approach has been adopted to allow a reasonable sampling process to occur when cluster sizes are small with respect to the GMRF's local neighbourhood support.

The first or likelihood term of the right hand side of equation 5.21 may be computed analytically via the approach outlined in Appendix C. Adopting the notation for the GMRF used in the section 2.2.3, equation 2.21, the resulting distribution is given by

$$\begin{aligned} p(\mathbf{y}_n \mid \mathbf{y}_b, \mu_c, \sigma_c, \boldsymbol{\theta}_c) \;\propto\; \frac{|\mathbf{R}_{nn}|}{\sigma_c^{N_n}} \exp\Big\{ -\frac{1}{2\sigma_c^2} \Big[& [\mathbf{y}_n - \boldsymbol{\mu}_c]^T \mathbf{R}_{nn} [\mathbf{y}_n - \boldsymbol{\mu}] \;+ \\ & 2[\mathbf{y}_n - \boldsymbol{\mu}_c]^T \mathbf{R}_{nb} [\mathbf{y}_b - \boldsymbol{\mu}_c] \;+ \\ & \big[\mathbf{R}_{nn}^{-1} \mathbf{R}_{nb} [\mathbf{y}_b - \boldsymbol{\mu}_c]\big]^T \mathbf{R}_{nb} [\mathbf{y}_b - \boldsymbol{\mu}_c] \Big] \Big\} \end{aligned} \tag{5.23}$$

where $\mathbf{y}_n$ is the N_n dimensional vector of observed gray-scale values for pixels comprising the cluster, $\mathbf{y}_b$ is the vector of observed gray-scale values for the bordering pixels

surrounding the cluster and the inverse covariance matrix for these pixels is given by $\boldsymbol{\Sigma}_c^{-1} = \begin{bmatrix} \mathbf{R}_{nn} & \mathbf{R}_{nb} \\ \mathbf{R}_{bn} & \mathbf{R}_{bb} \end{bmatrix}$, which comprises a sparse matrix whose non-zero elements are $\theta_c^{(\tau)}$ parameters. Because this expression is relatively complex, it may be necessary to approximate the likelihood term by the pseudo-likelihood:

$$\mathrm{PL}(\mathbf{y}_n \mid \mathbf{y}_b, \mu_c, \sigma_c, \boldsymbol{\theta}_c) \;\propto\; \frac{1}{\sigma_c^{N_n}} \exp\Big\{ -\frac{1}{2\sigma_c^2} \sum_{i \in \boldsymbol{\mathcal{C}}} \Big[(y_i - \mu_c)^2 - \sum_{j \in N_i} \theta_{(i-j)} (2 y_i y_j + y_j^2) \Big] \Big\} \tag{5.24}$$

where $\theta_{(i-j)} \equiv \theta_{(j-i)}$. All experimental results, presented later in section 5.7, were obtained using such an approximation and no discernible difference has been found between results generated using the two distributions.

As for the Isotropic case, the choice of statistic for generating the matrix of values comprising $\boldsymbol{\delta}$ is arbitrary. Recalling that the δ_{ij} parameters are intended to give an indication of distance between neighbouring pixels, it would appear desirable to include a measure of discrepancy in the spatial-frequency domain in addition to the difference in gray-scale distribution when considering the GMRF model. Such a measure will consist of a comparison between estimations of various spatial correlations at neighbouring sites within the image and to extract such information, Gabor filtering is perhaps the most obvious choice.

Gabor filters and their application to unsupervised texture segmentation were reviewed in section 3.4. By convolving the observed image with a set of suitable filters and then combining the output images (after suitable low-pass post-filtering) via some form of cost function, it is possible to generate a texture distance measure between neighbouring pixels.

To design a suitable set of Gabor functions it is desirable to match the filter banks half-peak coverage of the spatial-frequency domain with an estimate of the possible coverage of the particular GMRF model being used. The GMRF power spectral density function was shown in [97] to be

$$S(m,n) \;=\; \frac{\sigma^2}{1 - 2\sum_{(k,l) \in \boldsymbol{\mathcal{N}}_{GMRF}} \theta_{k,l} \cos\left(\frac{2\pi}{N}[mk + nl]\right)} \tag{5.25}$$

where $\boldsymbol{\mathcal{N}}_{GMRF}$ is the set of correlation displacements defining the GMRF. To calculate the half-peak coverage it is necessary to find the peak power spectral density. As the correlation parameters are relatively small and generally decrease in magnitude as their respective correlation distance increases, a crude approximation to the above equation would be to ignore all but the closest or nearest neighbour correlation parameters. This

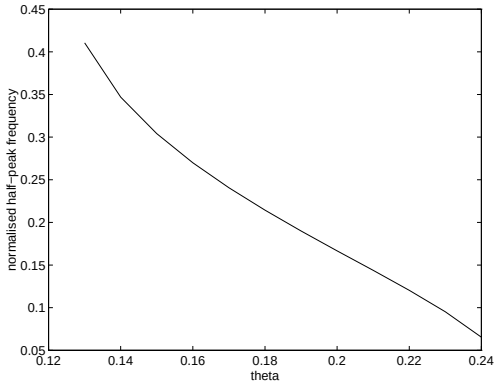


Figure 5.2: *Half-peak normalised frequency (i.e. frequency in cycles per image width = image width × normalised frequency) against the nearest neighbour correlation parameters, θ*

results in a peak power spectral density, $S_{MAX} = \frac{\sigma^2}{1-2(\theta_{0,1}+\theta_{1,0})}$. The locus of frequencies at the corresponding half-peak power spectral density is given by the equation,

$$\theta_{1,0}\cos\left(\frac{2\pi m}{N}\right) + \theta_{0,1}\cos\left(\frac{2\pi n}{N}\right) = 2(\theta_{1,0}+\theta_{0,1}) - \frac{1}{2} \tag{5.26}$$

and is therefore symmetrical about both frequency axes. If for simplicity both correlation parameters are assumed equal then it is possible to extrapolate the half-peak frequency from either of the two frequency axes, i.e. $\frac{m}{N} = \frac{1}{2\pi}\cos^{-1}\left(3-\frac{1}{2\theta}\right)$. From this equation, the relationship between half-peak frequency and the parameter θ, may be identified and is shown graphically in figure 5.2. In real images the nearest neighbour correlation parameters are rarely found to have a value less than 0.15, hence the maximum value of normalised frequency that should lie within the half-peak coverage of a Gabor Filter bank could reasonably be set to 0.3 cycles/image width[2].

Using this bandwidth requirement it is possible to specify a Gabor Filter bank by using the filter bandwidth equations 3.30. The filter outputs undergo post-filtering where they are convolved with a low-pass function to remove ripples induced by edges and boundary features. The resulting images are then divided into overlapping windows, over each of which, the absolute mean or 'texture energy' is computed. (These processes were discussed in more detail in section 3.4).

The spatial-frequency component of the δ_{ij} parameters is computed from the distance function formed by the difference in magnitudes of the vectors of output responses of neighbouring pixels: $d_{ij}^{(SF)} = \frac{1}{1+\sum_{f\in\mathcal{F}}(x_i^{(f)}-x_j^{(f)})^2}$, where $\mathbf{x}^{(f)}$ are the filter outputs, generated

by the set of filters $\boldsymbol{\mathcal{F}}$. As discussed earlier in section 5.4, when considering the Isotropic algorithm it is desirable to to normalise the distribution of the bond hyper-parameters for mean and standard deviation. The approach adopted here normalises spatial-frequency and Kolmogorov-Smirnov components separately, before they are summed to form a single set of hyper-parameters. The mean and variance statistics, μ_{gf} and σ_{gf}, are calculated for the $d_{ij}^{(SF)}$ distributions, allowing the normalisation to a specific mean and variance, $\mu_{\delta}^{(SF)}$ and $\sigma_{\delta}^{(SF)}$. The Gabor component of the hyper-parameters $\delta_{ij}^{(SF)}$ are then given by the expression $\delta_{ij}^{(SF)} = (d_{ij} - \mu_f)\frac{\sigma_{\delta}^{(SF)}}{\sigma_f} + \mu_{\delta}^{(SF)}$. The final δ_{ij} parameters may then be calculated by combining this value with a Kolmogorov-Smirnov component (previously derived as equation 5.12): $\delta_{ij} = \delta_{ij}^{(SF)} + \delta_{ij}^{(KS)}$.

5.5.2 Parameter Estimation

As for the Isotropic case, model parameters are sampled from their posterior distributions using similar methods to those of the previous chapter. Metropolis sampling is used to sample from an approximation to the posterior distribution. The approximation uses a form of pseudo-likelihood where clusters are assumed independent, rather than using the usual assumption where individual pixels are considered independent. Priors for μ_c and σ_c are kept strictly non-informative and are respectively uniform and Jeffrey's type distributions, made proper by restricting their range. For the $\theta_c^{(\tau)}$ parameters, conjugate priors are used, which comprise zero mean independent Gaussian random variables. The sole purpose of using conjugate priors in this case is to allow a direct comparison of results to be made with those obtained for the corresponding algorithm in the previous chapter. The resulting distribution is given by

$$\begin{aligned} p(\mu_c, \sigma_c, \boldsymbol{\theta}_c \mid \mathbf{y}) \quad &\approx \quad \frac{1}{\gamma} \prod_{C \in \boldsymbol{\mathcal{C}}_c} p(y_s, s \in C \mid \mu_c, \sigma_c, \boldsymbol{\theta}_c)\ p_r(\mu_c)\ p_r(\sigma_c)\ p_r(\boldsymbol{\theta}_c) \\ &\propto \quad \frac{1}{\sigma_c} \prod_{C \in \boldsymbol{\mathcal{C}}_c} p(y_s, s \in C \mid \mu_c, \sigma_c, \boldsymbol{\theta}_c) \end{aligned} \tag{5.27}$$

where $\boldsymbol{\mathcal{C}}_c$ is the set of clusters allocated to class c, N_c is the number of pixels allocated to class c, $p_r(\cdot)$ denotes a prior distribution and γ is an arbitrary normalising constant. The cluster likelihood term was given earlier as equation 5.24.

5.5.3 Sampling the Model Order

The process of proposing new model parameters from estimates generated from partitions of the observed data allows the acceptance rate of a reversible jump sampler to remain high, despite any increase in model complexity. This feature is illustrated by comparison of this GMRF algorithm with that of the simpler Isotropic MRF (described in section 5.4). If the region labelled by class selected to be split comprises a number of clusters, generated by the Partial Decoupling algorithm, then by selecting a pair from the largest N_c clusters, it is possible to generate two new sets of model parameters from the maximum likelihood estimates. For simplicity, due to the complex nature of the joint likelihood of a region or cluster modelled as a GMRF, the maximum likelihood estimates are approximated by the maximum pseudo-likelihood values. Hence, if the pseudo-likelihood for a GMRF region of n pixels is given by

$$\mathrm{PL}(y_1....y_n \mid \mu, \sigma, \boldsymbol{\theta}) \triangleq \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left[(y_i - \mu) - \sum_{\tau} \theta_\tau [(y_{i+\tau} - \mu) + (y_{i-\tau} - \mu)]\right]^2 \right\} \tag{5.28}$$

then, by expanding the exponent the equation may be re-written with the exponent in matrix form, where the elements of the matrices are sufficient statistics for the distribution:

$$\mathrm{PL}(y_1....y_n \mid \mu, \sigma, \boldsymbol{\theta}) \triangleq \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \left[\langle y_i^2 \rangle - 2\mu\langle y_i \rangle + \mu^2 - 2\boldsymbol{\theta}^T \mathbf{c} + \boldsymbol{\theta}^T \mathbf{D} \boldsymbol{\theta}\right] \right\} \tag{5.29}$$

where $\mathbf{c}$ is the vector of statistics $c_\tau = \sum_{i=1}^{n} (y_i - \mu)[(y_{i+\tau} - \mu) + (y_{i-\tau} - \mu)]$ and $\mathbf{D}$ is the matrix whose elements are the statistics $d_{\tau_1,\tau_2} = \sum_{i=1}^{n} [(y_{i+\tau_1} - \mu) + (y_{i-\tau_1} - \mu)][(y_{i+\tau_2} - \mu) + (y_{i-\tau_2} - \mu)]$.

The maximum likelihood estimates of the parameters may be found by setting their partial derivatives to zero. Hence, if pixels neighbouring the region are assumed to have a similar mean, then setting $\frac{\partial PL(y_1....y_n|\mu,\sigma,\boldsymbol{\theta})}{\partial \mu} = 0$ yields $\hat{\mu} = \langle y_i \rangle$. Equating $\frac{\partial PL(y_1....y_n|\mu,\sigma,\boldsymbol{\theta})}{\partial \theta_{\tau_k}} = 0$, $\forall k$ and recalling that $\mathbf{D}$ is symmetric produces a system of linear simultaneous equations in $\hat{\theta}_{\tau_k}$: $\mathbf{D}\hat{\boldsymbol{\theta}} = \mathbf{c}$, whose solution is $\hat{\boldsymbol{\theta}} = \mathbf{D}^{-1}\mathbf{c}$. Although this estimate does not guarantee stability, the parameter estimates are being used for segmentation and thus computational simplicity rather than stability is of prime concern. Similarly, setting $\frac{\partial PL(y_1....y_n|\mu,\sigma,\boldsymbol{\theta})}{\partial \sigma} = 0$ gives the maximum pseudo likelihood estimate $\hat{\sigma}^2 = \frac{1}{n}\left[\langle y_i^2 \rangle - 2\mu\langle y_i \rangle + \mu^2 - 2\hat{\boldsymbol{\theta}}^T \mathbf{c} + \hat{\boldsymbol{\theta}}^T \mathbf{D} \hat{\boldsymbol{\theta}}\right]$.

To ensure the reverse transition is possible, a small random perturbation must be added to each of these estimates:

$$\begin{array}{lllllll} \mu_{c1} & = & \hat{\mu}_{c1} + e_{m1}, & & \mu_{c2} & = & \hat{\mu}_{c2} + e_{m2}, \\ \sigma_{c1} & = & \hat{\sigma}_{c1} + e_{s1}, & & \sigma_{c2} & = & \hat{\sigma}_{c2} + e_{s2}, \\ \theta_{c1}^{(\tau_k)} & = & \hat{\theta}_{c1}^{(\tau_k)} + e_{t1}^{(k)}, & & \theta_{c2}^{(\tau_k)} & = & \hat{\theta}_{c1}^{(\tau_k)} + e_{t2}^{(k)}, \quad \forall k \end{array} \tag{5.30}$$

where e_{m1}, e_{m2}, e_{s1}, e_{s2}, $e_{t1}^{(k)}$ and $e_{t2}^{(k)}$ are perturbation random variables, drawn from zero mean i.i.d. Gaussian and shifted Gamma distributions. The probability of proposing the new pair of parameter vectors is therefore

$$\begin{aligned} q(\boldsymbol{\phi}_{c1},\boldsymbol{\phi}_{c2} \mid \mathbf{u},\mathbf{y}) \;=\; & \frac{1}{N_c(N_c-1)} \sum_{i=1}^{N_c} \sum_{j=1:j\neq i}^{N_c} p(e_{m1} = \mu_{c1} - \hat{\mu}_i)\; p(e_{m2} = \mu_{c2} - \hat{\mu}_j) \\ & \times p(e_{s1} = \sigma_{c1} - \hat{\sigma}_i)\; p(e_{s2} = \sigma_{c2} - \hat{\sigma}_j) \prod_k p(e_{t1}^{(k)} = \theta_{c1}^{(k)} - \hat{\theta}_i^{(k)})\; p(e_{t1}^{(k)} = \theta_{c2}^{(k)} - \hat{\theta}_j^{(k)}) \end{aligned} \tag{5.31}$$

where N_c is the number of pixels allocated to class c. The reverse move entails the combining of two sets of parameters into one. Two mechanisms for this process appear intuitively profitable: firstly, generation of the new parameters by maintaining the central moments of the model in a similar fashion to that used in the Isotropic case, or secondly, by using the observed data that would be labelled by the new combined class to estimate new model parameters using the maximum pseudo-likelihood technique outlined above.

The first of these processes uses identical equations to the Isotropic case (i.e. equations 5.16 to 5.18) to generate the μ_c and σ_c parameters. To generate the combined correlation parameter vector $\boldsymbol{\theta}_c$, an *ad hoc* averaging equation may be used: $\theta_c^{(k)} = \frac{N_{c1}\theta_{c1}^{(k)} + N_{c2}\theta_{c2}^{(k)}}{N_{c1}+N_{c2}} + a_t^{(k)}$, since no obvious system of central moment conservation exists.

The second process uses expressions similar to those used when calculating the splitting move:

$$\begin{array}{lllll} \mu_c & = & \hat{\mu}_c + a_m & & \\ \sigma_c & = & \hat{\sigma}_c + a_s & & \\ \theta_c^{(\tau_k)} & = & \hat{\theta}_c^{(\tau_k)} + a_t^{(k)}, & \forall k & \end{array} \tag{5.32}$$

Both systems of equations achieve similar results, but have the slightly differing proposal probabilities: $q(\boldsymbol{\phi}_c \mid \mathbf{x},\mathbf{y}) = q(\boldsymbol{\phi}_c \mid \boldsymbol{\phi}_{c1},\boldsymbol{\phi}_{c2},\mathbf{x}) = p(a_m)\; p(a_s) \prod_k p(a_t^{(k)})$ and $q(\boldsymbol{\phi}_c \mid \mathbf{x},\mathbf{y}) = \frac{1}{N_c}\sum_{i=1}^{N_c} p(a_m = \mu_c - \hat{\mu}_i)\; p(a_s = \sigma_c - \hat{\sigma}_i) \prod_k p(a_t^{(k)} = \theta_c^{(k)} - \hat{\theta}_i^{(k)})$ respectively. Both also give the necessary bijection between auxiliary parameter spaces:

$[\mu_c, \sigma_c, \boldsymbol{\theta}_c, e_{m1}, e_{m2}, e_{s1}, e_{s2}, \mathbf{e}_{t1}, \mathbf{e}_{t2}] \leftrightarrow [\mu_{c1}, \mu_{c2}, \sigma_{c1}, \ \sigma_{c2}, \boldsymbol{\theta}_{c1}, \boldsymbol{\theta}_{c2}, a_s, a_m, \mathbf{a}_t]$. The resulting Jacobian determinants are both unity.

The re-segmentation algorithm is identical to that used in the Isotropic case (see section 5.4) and so the probability of proposed re-segmentation $q(\mathbf{x}, \mathbf{x}^+)$, is given by equation 5.19. Hence, the acceptance ratio for the proposed GMRF is

$$\min\left[1, \ \frac{\pi(\mathbf{x}^+, \boldsymbol{\psi}^+ \mid \mathbf{y})}{\pi(\mathbf{x}, \boldsymbol{\psi} \mid \mathbf{y})} \frac{q(\text{merge})}{q(\text{split})} \frac{1}{k+1} \frac{q(c1, c2)}{q(c)} \frac{q(\boldsymbol{\phi}_c \mid \boldsymbol{\phi}_{c1}, \boldsymbol{\phi}_{c2}, \mathbf{x})}{q(\boldsymbol{\phi}_{c1}, \boldsymbol{\phi}_{c2} \mid \mathbf{u}, \mathbf{y})} \frac{1}{q(\mathbf{x}, \mathbf{x}^+)}\right] \tag{5.33}$$

5.6 Low-level Line Processes

The line process of Geman & Geman[39] comprises a binary Markov process (see section 2.2.2). It is defined by a set of potential functions acting on corresponding cliques which comprise a neighbourhood. By the Hammersley-Clifford theorem[9] the joint distribution for such a process is Gibbsian. The juxtaposition of the line sites with respect to the underlying Markov Random Field pixel sites is identical to that of the auxiliary bond variables used in the partial-decoupling algorithm, as discussed in the previous sections. However, their interaction with the underlying Markov Random Field is quite different.

The joint posterior density for a Markov Random Field and associated line process may be expressed by the equation

$$\pi(\mathbf{x}, \mathbf{l} \mid \boldsymbol{\Psi}, \mathbf{y}) \ \propto \ \exp\Big\{ \sum_{i \in S} -\frac{1}{2}\Big(\frac{y_i - \mu_{x_i}}{\sigma_{x_i}}\Big)^2 + \sum_{i \in S} \sum_{j \in \eta_i} (1 - l_{ij}) \beta V_c(x_i, x_j) + V_l(l_{ij}, l_{\eta_{ij}}) \Big\} \tag{5.34}$$

Thus, if the line element situated between pixels at sites i and j takes the value of 1, then the two pixels become mutually independent. The line process potential functions $V_l(l_{ij}, l_{\eta_{ij}})$ may be arbitrarily defined to prefer particular arrangements of line process elements. Typically, straight line segments are favoured, while corners, line endings and isolated elements are rejected with increasing strength.

To date, when considering unsupervised segmentation, the incorporation of line processes into the model has been largely precluded simply because the typical two-step algorithms, described in chapter 3 are unsuitable for fitting observed data to these more

complex models. However, the unsupervised partial decoupling algorithms of this chapter may be used to optimise a line process model: this possibility will be now be examined in more detail[2]. To implement such an algorithm, the auxiliary bond variables must remain uniformly and independently distributed. The joint distribution for the bond variables, the underlying MRF and the line process is therefore given by

$$\begin{aligned}\pi(\mathbf{x},\mathbf{l},\mathbf{u}\mid\boldsymbol{\psi},\mathbf{y}) &= p(\mathbf{u}\mid\mathbf{x})\;\pi(\mathbf{x},\mathbf{l}\mid\boldsymbol{\psi},\mathbf{y}) \\ &\propto \exp\Big\{\sum_{i\in S}-\frac{1}{2}\Big(\frac{y_i-\mu_{x_i}}{\sigma_{x_i}}\Big)^2+\sum_{j\in\eta_i}(1-l_{ij}-\delta_{ij})\beta V_c(x_i,x_j)+V_l(l_{ij},l_{\eta_{ij}})\Big\} \\ &\quad\times\prod_{i,j\in S}\mathrm{I}\big[0\le u_{ij}\le\exp\{\delta_{ij}\beta_{ij}V_c(x_i,x_j)\}\big]\end{aligned} \tag{5.35}$$

The marginal of this distribution, removing the auxiliary bond variables is quite clearly the posterior distribution for the Markov Random Field and line process,
i.e. $\int_{\mathbf{U}}\pi(\mathbf{x},\mathbf{l},\mathbf{u}\mid\boldsymbol{\psi},\mathbf{y})d\mathbf{u}=\pi(\mathbf{x},\mathbf{l}\mid\boldsymbol{\psi},\mathbf{y})$.

The particular optimisation problem of interest here is the joint MAP estimation of the hidden label variables, the line processes, the model parameters and the model order:

$$\begin{aligned}&\hat{\mathbf{x}}_{MAP},\;\hat{\mathbf{l}}_{MAP},\;\hat{\boldsymbol{\theta}}_{MAP},\;\hat{k}_{MAP}\;= \\ &\qquad\arg\max_{\mathbf{x}\in\mathbf{X}_{(\mathbf{k})},\;\mathbf{l}\in\mathbf{L},\;\boldsymbol{\theta}_{(k)}\in\boldsymbol{\Theta}_{(k)},\;k\in\mathbf{K}}\;p(k)\;p(\boldsymbol{\theta}_{(k)})\;p(\mathbf{x}\mid\boldsymbol{\theta}^{(X)}_{(k)},\mathbf{l})\;p(\mathbf{l}\mid\boldsymbol{\theta}^{(L)}) \\ &\qquad\qquad\times\prod_{c\in\boldsymbol{\Lambda}_{(k)}}p\big(\{\mathbf{y}_s,s:x_s=c\}\mid\boldsymbol{\theta}^{(Y)}_{(k)_c}\big)\end{aligned} \tag{5.36}$$

Knowledge of this joint distribution facilitates sampling variables from their appropriate conditional distributions when available in closed form and when not, the full joint distribution may be used. Hence adopting the previous notation, the conditional distribution for the hidden labels of cluster $\boldsymbol{\mathcal{C}}$ given the current bond configuration and line process is

$$\begin{aligned}p\big(\{x_i=c,i\in\boldsymbol{\mathcal{C}}\}\mid\mathbf{u},\boldsymbol{\delta},\beta,\mu_k,\sigma_k,\mathbf{l},\mathbf{y}\big)\;&\propto\;p\big(\{y_i,i\in\boldsymbol{\mathcal{C}}\}\mid\{y_j,j\in\partial\boldsymbol{\mathcal{C}}\},\mu_k,\sigma_k\big) \\ &\quad\times\;p\big(\{x_i=k,i\in\boldsymbol{\mathcal{C}}\}\mid\mathbf{u},\boldsymbol{\delta},\beta,\mathbf{l}\big)\end{aligned} \tag{5.37}$$

where the first, likelihood term is dependent on the particular model, e.g. Isotropic or GMRF and the second term is given by

$$p\big(\{x_i=c,i\in\boldsymbol{\mathcal{C}}\}\mid\mathbf{u},\boldsymbol{\delta},\beta\big)\;\propto\;\exp\Big\{\beta\sum_{i\in\boldsymbol{\mathcal{C}}}\sum_{j\in\partial\boldsymbol{\mathcal{C}}}(1-l_{ij}-\delta_{ij})V(x_i,x_j)\Big\} \tag{5.38}$$

[2]The work included in this section was presented at the 'International Symposium on Optical Science, Engineering and Instrumentation', SPIE's 43rd Annual Meeting in San Diego, California and was published in the conference proceedings [3].

Finally, the conditional distribution for $\mathbf{l}$ given the bond configuration and hidden labels is

$$p(l_{ij} \mid \mathbf{u}, \mathbf{x}, \mathbf{l}_{\mathbf{\Omega}_L \setminus ij}) \;\propto\; \exp\left\{ - l_{ij}\beta V(x_i, x_j) + U(l_{ij} \mid \mathbf{l}_{\mathbf{\Omega}_L \setminus ij}) \right\} \tag{5.39}$$

where $\mathbf{\Omega}_L$ is the lattice on which the line process is defined and $U(l_{ij} \mid \mathbf{l}_{\mathbf{\Omega}_L \setminus ij})$ is the contribution of the prior term to the conditional energy function.

Using the above conditional equations, it is possible to construct an optimisation process by modifying the iterative algorithm outlined in section 5.3: an extra step must be incorporated to update the line process.

5.7 Results

The Swendsen-Wang and partial decoupling algorithms allow a sampler to mix more rapidly at the critical temperature. The Gibbs sampler is poorer mixing around the critical temperature and hence, experiments in the previous chapter required an annealing schedule that was slowly lowered from a starting temperature well above this temperature. The improvement in sampling associated with partial decoupling allows the annealing schedule to begin at a lower temperature, thus reducing the time wasted when effectively sampling from a distribution with different characteristics.

The annealing schedules for all experiments are linear and are given by,

$$T_t \;=\; \frac{(t-1)}{n}(T_{max} - T_{min}) + T_{min} \tag{5.40}$$

where t is the iteration number and n is the maximum number of iterations. T_{max} and T_{min} are the respective starting and finishing temperatures of the annealing schedule.

The Isotropic unsupervised segmentation algorithm described in section 5.4, has been tested on a variety of noisy gray-scale images. Figure 5.3 shows both the MAP segmentation and marginal distribution for the number of states k, for a simple four state image. Here the MRF hyper-parameter β was set to 1.2 *a priori*, the bond hyper-parameter collective mean μ_δ to 0.30 and the associated variance σ_δ to 0.30. To obtain the marginal 10,000 samples were drawn from the posterior distribution.

A further, more complex computer synthesised mosaic, shown in Figure 5.4, which was also used to test the earlier Isotropic unsupervised segmentation algorithm, presented

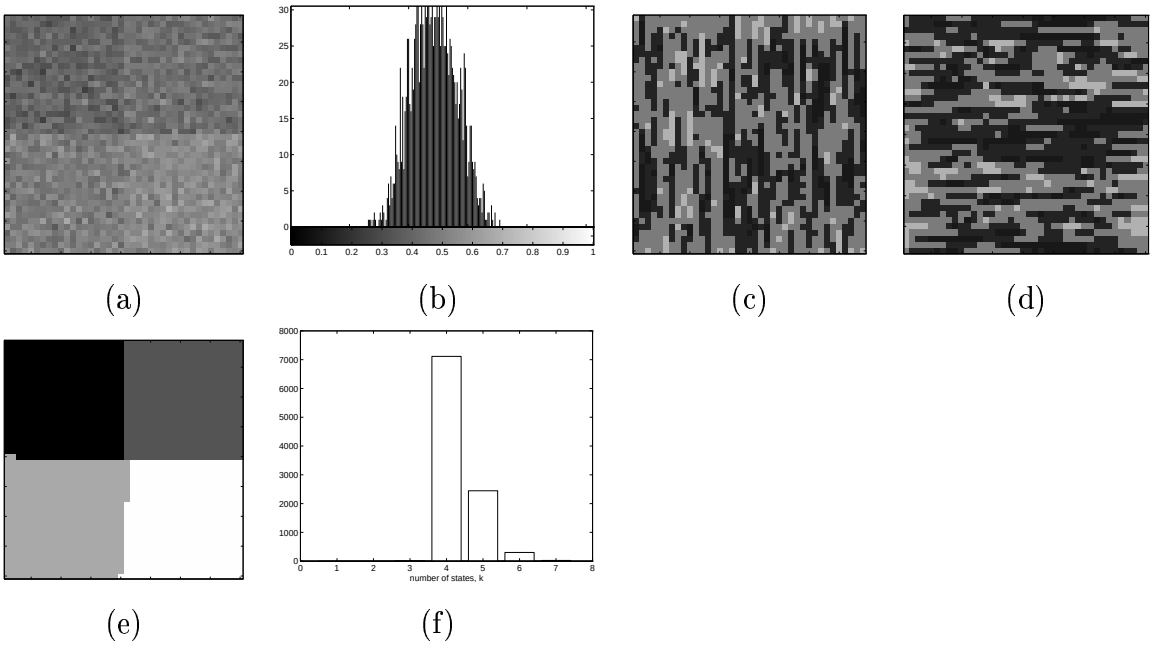


Figure 5.3: *Estimation of Marginal Distribution for a four state image: (a) observed image, (b) gray-scale histogram of the observed image, (c) horizontal bond hyper-parameters, (d) vertical bond hyper-parameters, (e) MAP segmentation, (f) marginal distribution for the number of states k calculated over 10,000 samples.*

in section 4.5. To allow a comparison with these previous results, shown in Figure 4.2, parameter values are set to those of the previous chapter, i.e. the MRF hyper-parameter is set according to $\beta\mu_\delta = \beta_{PREVIOUS}$. The horizontal and vertical priors for the auxiliary bond variable hyper-parameters are also displayed in Figure 5.4, with the light coloured regions corresponding to areas homogeneity between pixels, generating high bond prior probabilities. Estimations of the underlying MRF are shown above their associated bond-maps and the difference in sampling algorithm, when compared to the previous chapter is perhaps best illustrated after 200 iterations, where the clustering associated with the partial decoupling algorithm is clearly visible. The final bond-map effectively comprises a negative of an image containing the borders between regions, with most internal pixels being bonded. The new algorithm's convergence appears quicker and its final segmentation more accurate than that of its predecessor and so for this image represents a large improvement.

The new GMRF unsupervised segmentation may also be compared with that of the previous chapter by comparison of Figure 4.5 with Figure 5.5. Again, to facilitate such a comparison, prior parameter values are set to those of the previous chapter, hence, the new

parameters	class	μ	σ	θ_1	θ_2	θ_3	θ_4
actual	1	0.50	0.10	-0.10	0.20	0.20	0.20
	2	0.50	0.10	0.34	0.32	-0.21	-0.25
	3	0.50	0.10	0.20	0.20	-0.20	0.20
	4	0.50	0.10	0.20	0.20	0.20	-0.20
estimated	1	0.503354	0.104674	-0.124109	0.184704	0.236364	0.186041
	2	0.503244	0.101905	0.347301	0.344144	-0.223274	-0.23885
	3	0.509906	0.103892	0.207443	0.207297	-0.215471	0.19872
	4	0.505321	0.105929	0.191792	0.157124	0.225666	-0.17101

Table 5.1: *Actual and estimated parameters for the partial decoupling GMRF segmentation.*

MRF hyper-parameter is set according to, $\beta(\mu_\delta^{(SF)}+\mu_\delta^{(KS)}) = \beta_{PREVIOUS}$. The textures in this synthesised image have identical means and variances and are thus only distinguishable by the their correlation parameters. The threefold increase in convergence rate for the new algorithm is dramatic and may be partly attributable to the lower annealing starting temperature made possible by the use of partial decoupling. The parameter estimates obtained while segmenting the synthesised textures are given in Table 5.1. These represent an improvement over those obtained in the previous chapter (see Table 4.1) which may be attributed to the more accurate pseudo-likelihood approximation, as was described in section 5.5.2.

Figures 5.6 and 5.7 show the results of an unsupervised segmentation of a mosaic of four Brodatz textures [16]. The GMRF model used in this example is defined over a larger neighbourhood (type $n = 8$, see Chellappa[54]) than that used in the synthetic example. Such an increase in model complexity (the number of correlation parameters to be estimated has increased from 4 to 22, for each texture class) is made possible with the improvement in convergence attributable to the partial decoupling algorithm.

Figure 5.8 illustrates the application of GMRF unsupervised segmentation algorithm to satellite image segmentation. Again a $n = 8$ neighbourhood structure is used in the GMRF model for each region. The Potts model hyper-parameter was set *a priori* to $\beta = 1.2$.

The algorithm demonstrates its capability by differentiating between areas of similar gray-level, for example it discriminates between the lagoon on the top left-hand side of the image and the built up areas towards the bottom right. However the differential

between open sea and scrub land proves too small, since the gray-scale distributions are virtually identical and there is little textural content, thus the algorithm misclassifies these as identical regions. More encouragingly, smaller details are accurately located, for example the beach and yachting marina are clearly segmented.

Results are also presented demonstrating the unsupervised segmentation of an hierarchical Isotropic MRF with line process. The line process was formulated simply to favour line segments only. Letting L denotes the presence of a line element and $\cdot$ its absence, then actual potentials were defined for the central site of three adjacent elements such that: $V_l(LLL) = 0.2$; $V_l(LL\cdot) = 1.5$; $V_l(\cdot L\cdot) = 4.0$; $V_l(L \cdot L) = 3.0$; $V_l(L \cdot \cdot) = 1.0$, and $V_l(\cdot \cdot \cdot) = 0.0$. Figure 5.9 shows the results of optimising such a model on a 100×100 image containing six gray-levels. As would be expected for such a simple image the final line process forms practically a negative of the eventual bond configuration.

Figure 5.10 shows the fitting of an identical line process model to an image of a house. The image is identical to that used to test the earlier Isotropic segmentation algorithm of chapter 4, shown in Figure 4.4. The auxiliary bond variables and MRF hyper-parameter were chosen to allow a direct comparison between the segmentations produced using the two models. Due to the reduction in cost associated with the incorporation of a line process the estimated number of states has increased from 5 to 11. More detail has been segmented using the line process (particularly note the window and drainpipe areas) however the roof appears to have become over segmented.

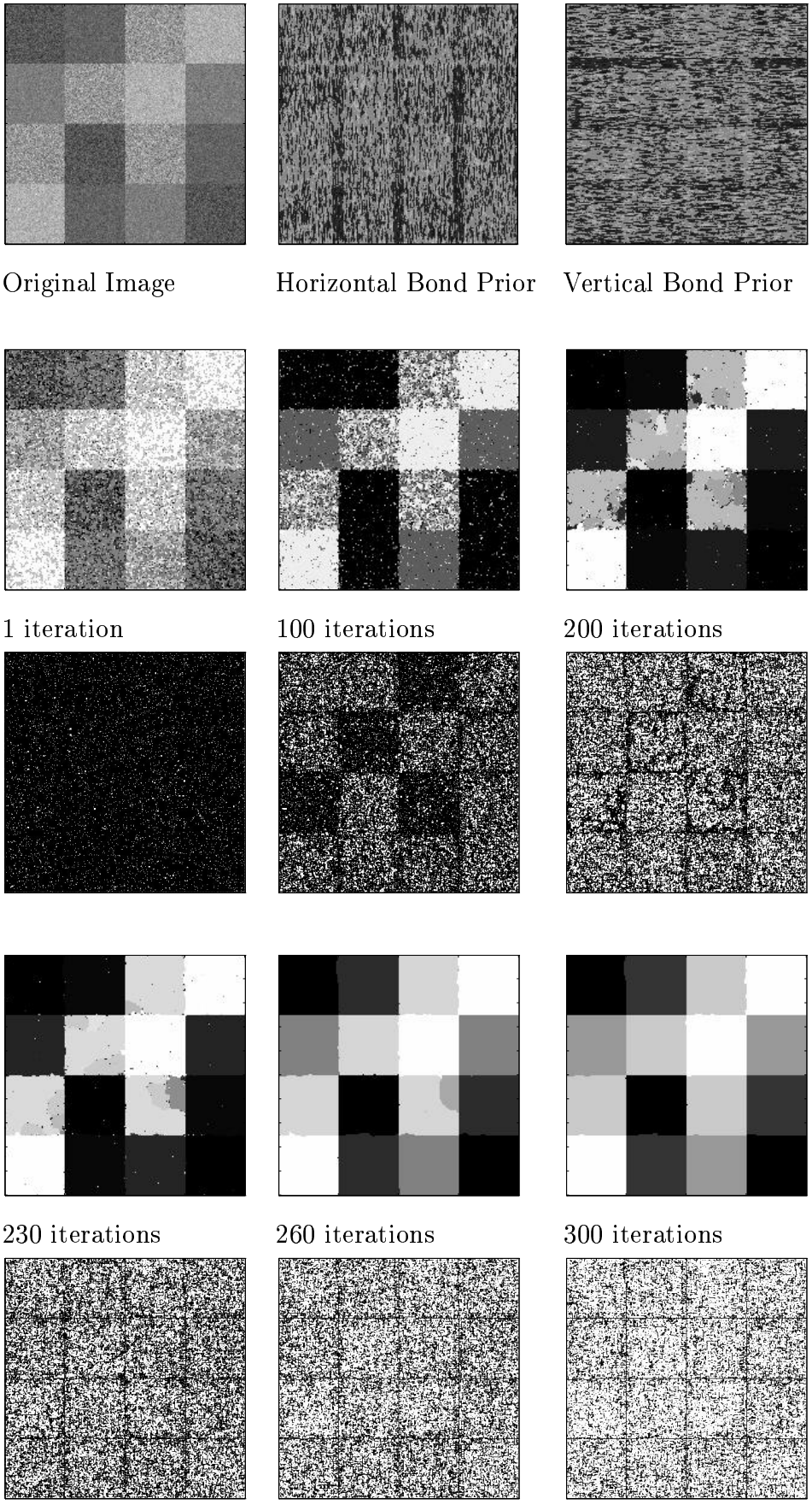


Figure 5.4: *Isotropic MRF Segmentation experiment using partial decoupling. Segmentations are shown above their associated bond-maps at various stages of annealing*

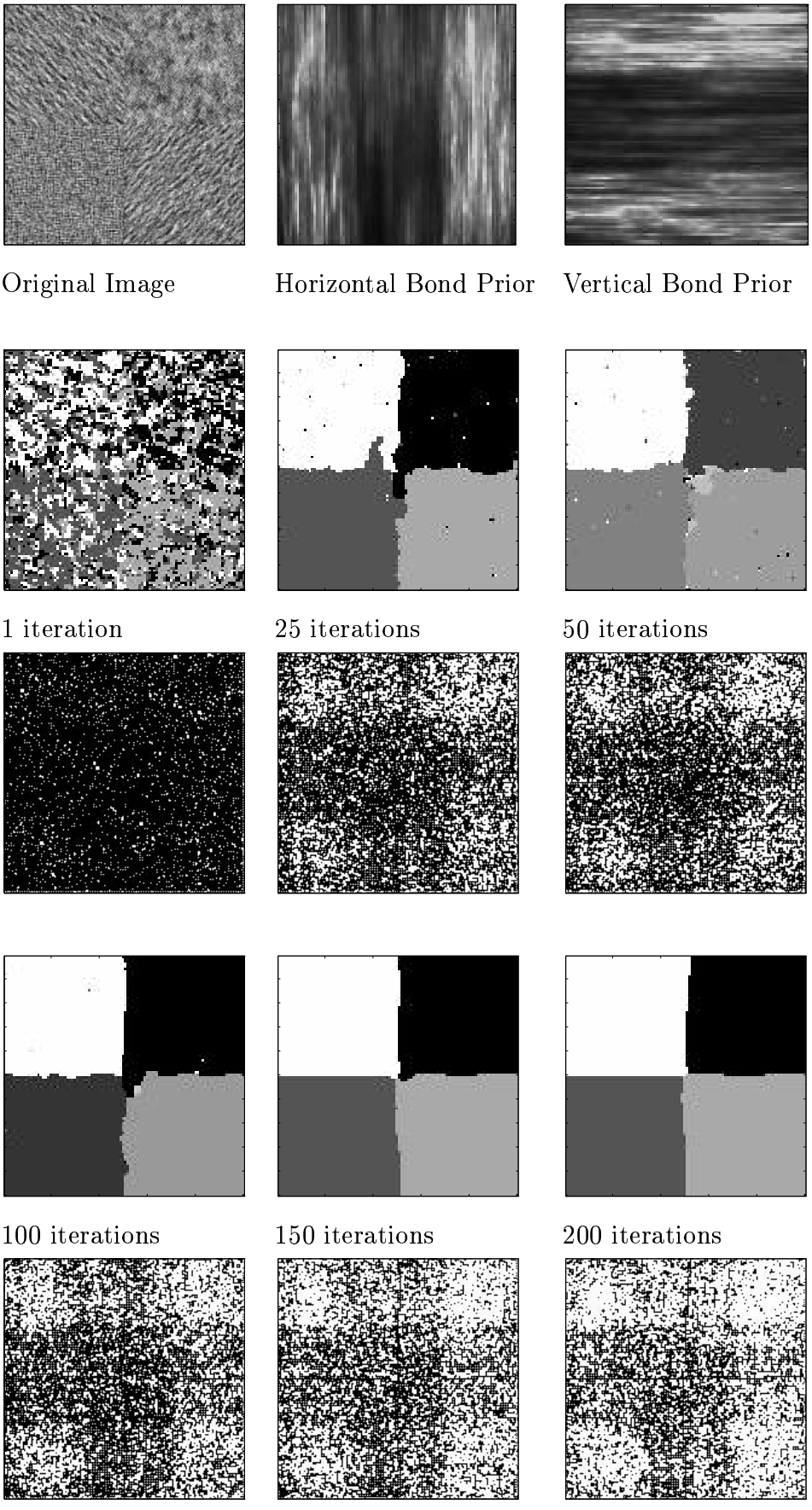


Figure 5.5: *Synthesised Texture Segmentation experiment. Segmentations with their associated bond-map at various stages of the unsupervised partial decoupling algorithm*

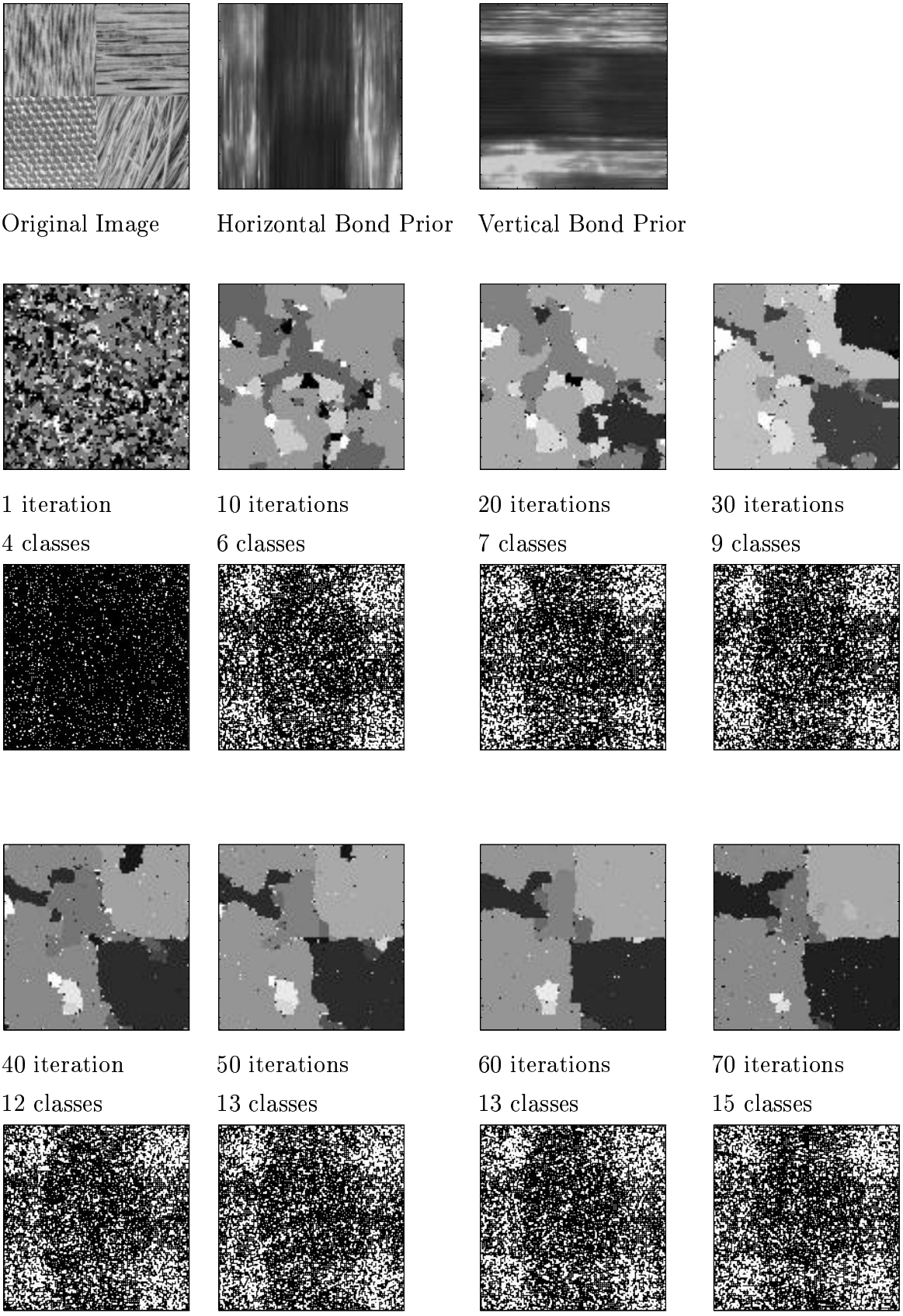


Figure 5.6: *The first 70 iterations showing particular texture segmentations and their associated bond-maps obtained using the unsupervised partial decoupling algorithm applied to a mosaic of four Brodatz textures*

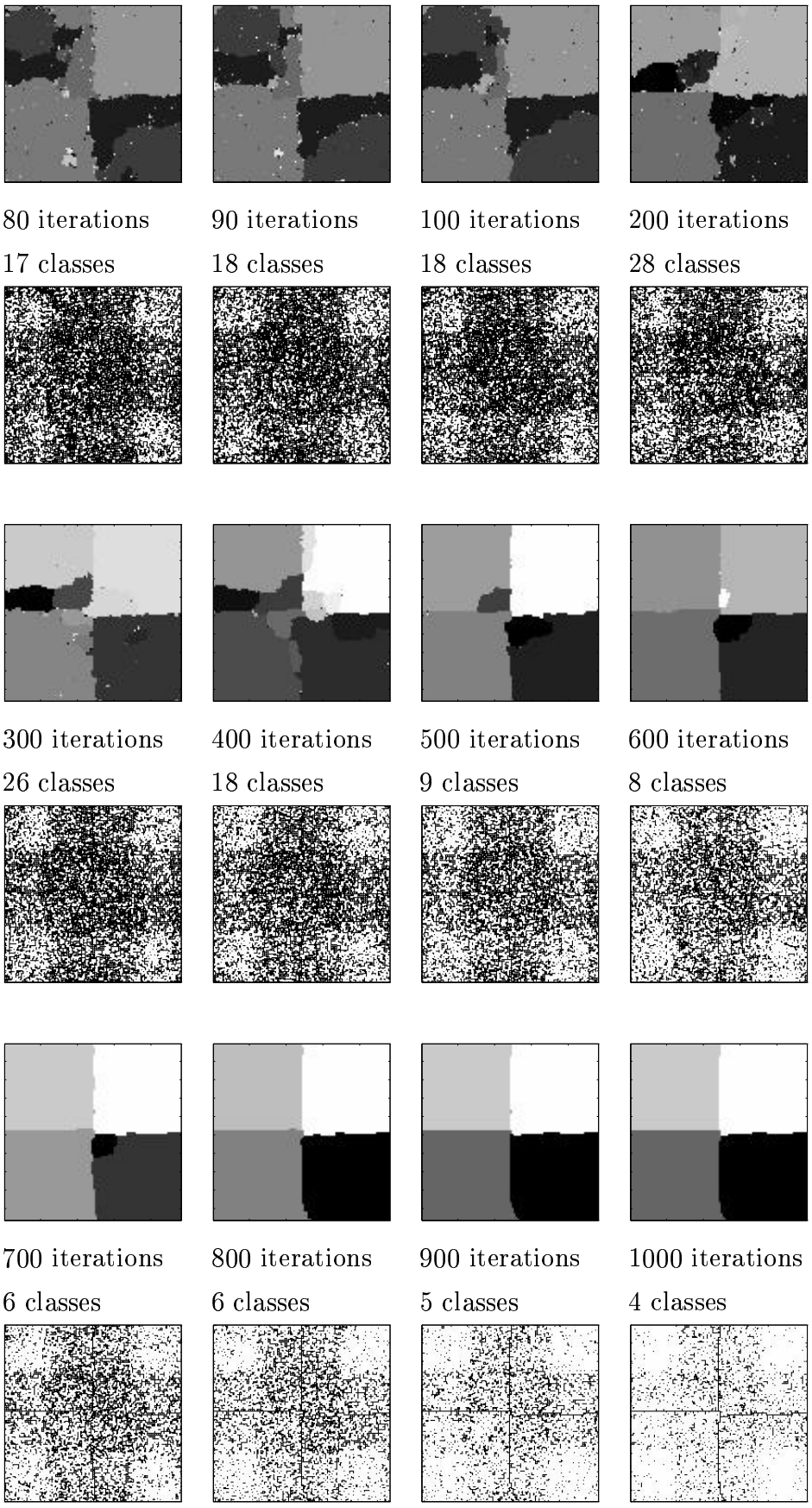
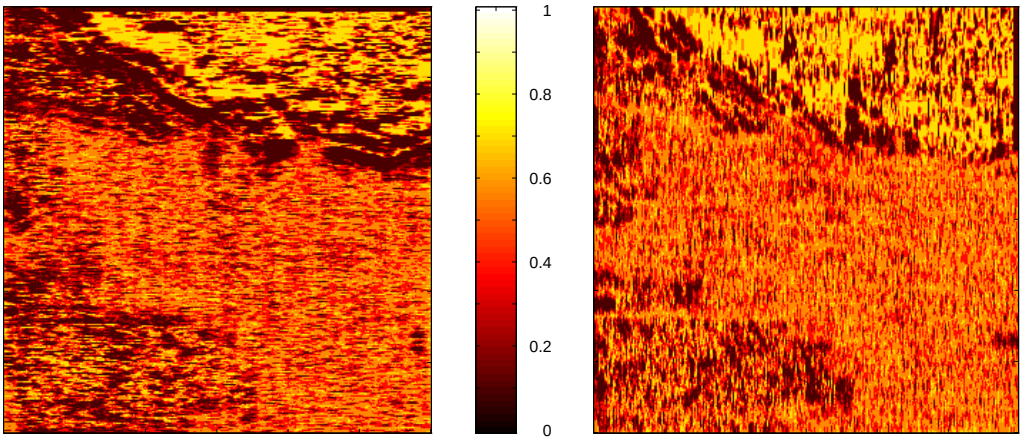


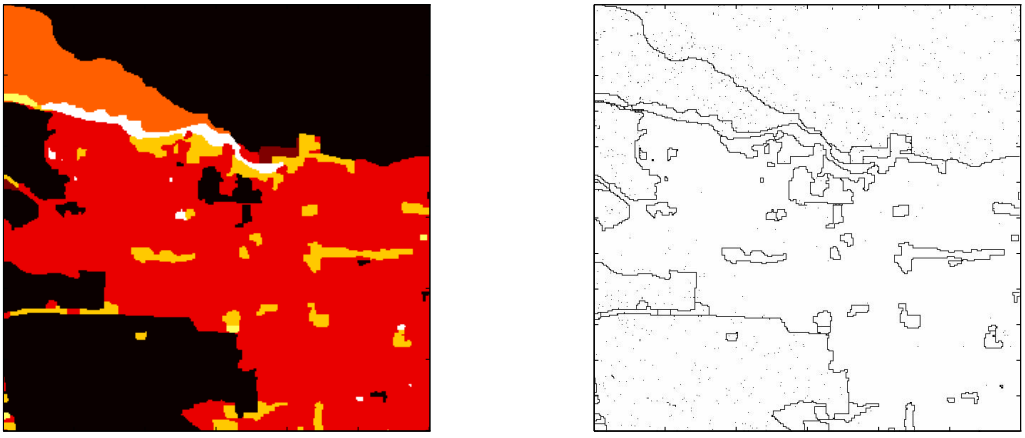
Figure 5.7: *The final 830 iterations showing particular texture segmentations and their associated bond-maps obtained using the unsupervised partial decoupling algorithm applied to a mosaic of four Brodatz textures*

Original Image



Horizontal Bond Hyper-Parameters

Vertical Bond Hyper-Parameters



Segmentation

Bond Configuration

Figure 5.8: *Results of a 200 iteration unsupervised segmentation algorithm applied to a* 300×300 *satellite image of a coastal region in British Columbia.*

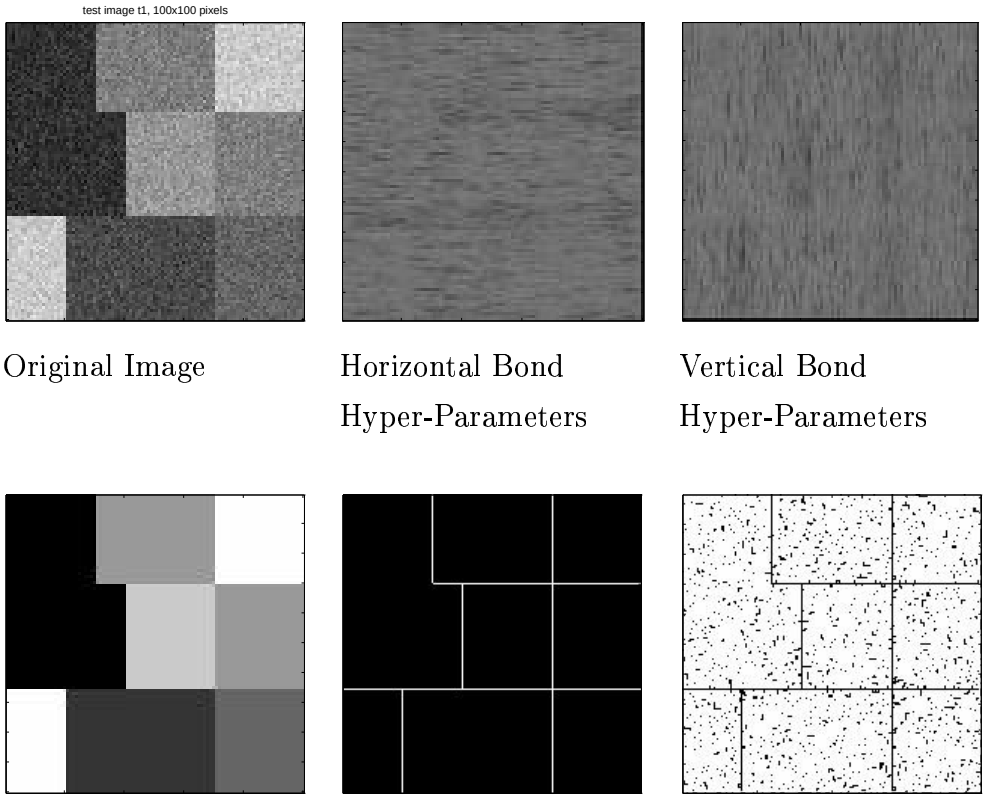
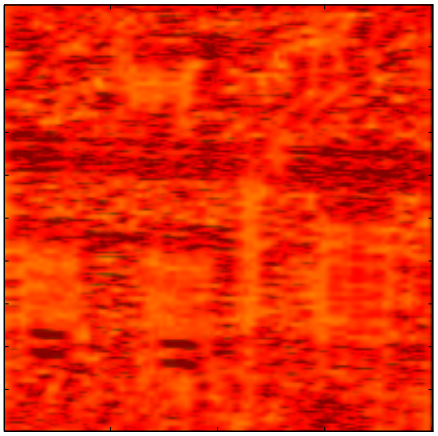


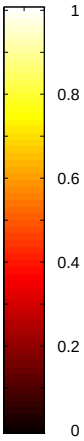
Figure 5.9: *200 iteration line process experiment.*

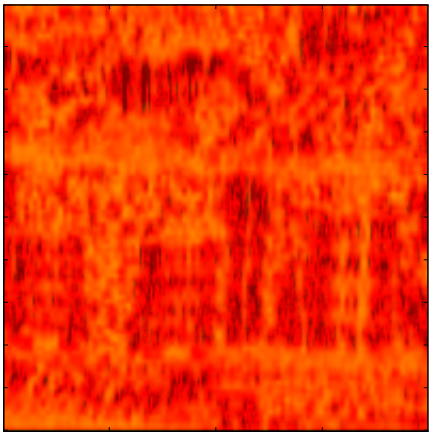
Original Image

Segmentation into 11 states



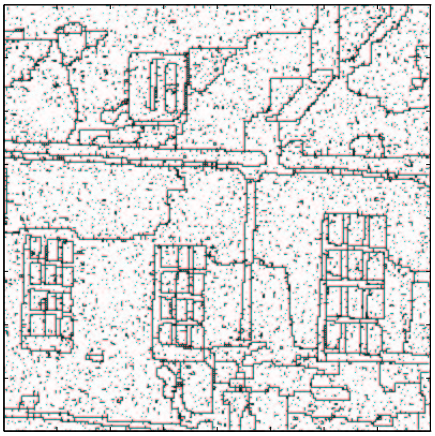
Vertical Bond
Hyper-Parameters





Horizontal Bond
Hyper-Parameters

Line Process



Bond Configuration

Figure 5.10: *200 iteration line process experiment using an image of a house.*

5.8 Conclusion

In this chapter, auxiliary variable techniques have been used to improve upon the unsupervised segmentation algorithms introduced in the previous chapter. In particular, convergence has been improved through two procedures: firstly through the use of the partial decoupling algorithm to accelerate mixing at and below the systems critical temperature, and secondly, through generating reversible jump proposals intelligently, thus increasing the acceptance rate.

The application of the partial decoupling algorithm, with the implied specification of hyper-parameters for the auxiliary bond variables allows the use of *ad hoc* procedures to improve convergence. Specifically, the hyper-parameters provide a mechanism by which prior knowledge pertaining to the local homogeneity of the observed data may be incorporated into the optimisation process without affecting the final segmentation. The texture features reviewed in chapter 3 are prime candidates to form the basis for the generation of these hyper-parameters. Here, a bank of Gabor filters has been applied for this purpose, however any of the features, for example those derived from cooccurrence matrices or even the model itself, may be suitable.

The results presented demonstrate the advance produced by the incorporation of these processes. For example, the segmentation of an image composed of Brodatz textures requires GMRF models of significant support. The use of composition sampling in the previous chapter, although making possible the unsupervised segmentation of hierarchical models comprising small neighbourhood GMRF's was not an efficient enough process to solve this problem, mainly due to the difficulty in deriving good enough proposals to maintain a reasonable level of acceptance in the reversible jump sampler. The use of partial-decoupling in the algorithms of this chapter, through the introduction of clusters on a smaller scale than the segmented regions has formed the basis for an intuitive process of proposal generation, thus improving acceptance rates and consequently the algorithm's convergence rate.

Conclusions 6

The development of unsupervised image segmentation algorithms based upon hierarchical Markov Random Field (MRF) image models has been the central theme of this dissertation. Results have been presented demonstrating their effectiveness in segmenting a variety of computer generated and real world images.

The algorithms may be viewed as an attempt to provide the functionality of the low-level elements of a basic vision process. It is hoped that such algorithms, when combined with higher level models and procedures, for example shape or object recognition, will facilitate the future realisation of an automatic vision system. However, computational limitations presently dictate that these algorithms may be usefully applied to only a subset of vision problems, for example: remote sensing applications, X-ray and nuclear magnetic resonance (NMR) image segmentation, and segmentation of synthetic aperture radar (SAR) images for crop discrimination.

In chapter 2 the supervised, semi-unsupervised and fully unsupervised image segmentation problems were formulated in terms of an hierarchical image model. The hierarchical model is Bayesian and is comprised of MRF's: at the low level these model the observed texture regions that make up the image; at the higher level they regularise the segmentation. Textures are modelled by Gaussian MRF's and the regularising function takes the form of a Potts model. MRF's were introduced generally in section 2.2, before both the Potts model and GMRF were described in more detail, with particular emphasis on how the Potts model's characteristics vary with temperature. This is of prime concern when optimising using the simulated annealing algorithm.

A review of the various techniques used to achieve segmentation, adopted throughout the literature, form the remainder of chapter 2. Algorithms such as Gibbs sampling, Simulated Annealing, Iterative Conditional Modes (ICM), Expectation-Maximisation (EM), Iterative Conditional Expectation (ICE), Partial-Optimal Solutions, Mean Field Theory, soft decision EM, and Information Criteria were all discussed. However, despite many

encouraging results these algorithms all fail to achieve the goal of truly unsupervised segmentation. It was also observed that most require extensive computational resources.

A parallel area of research, reviewed in chapter 3, makes use of various feature spaces to facilitate segmentation through clustering or further model fitting algorithms. The features considered include those simply derived from the models of chapter 2, functions of cooccurrence matrices, spatial residuals and features generated in the spatial frequency domain. These algorithms, although requiring the precise specification of various thresholding parameters and despite giving worse localisation of image boundaries due to the use of windowing, consume considerably less computational resources than their model based counterparts.

Having reviewed much of the image segmentation literature, the primary motivation behind the work reported on in this dissertation was established. Due to the difficulties in achieving unsupervised and accurate segmentations using the feature based approaches, a truly unsupervised segmentation algorithm based upon the optimisation of an image model became the end goal of the research. However, due to the enormous computation overheads the incorporation of features to speed up the optimisation process was a secondary objective.

The first algorithms were presented in chapter 4. The reversible jump Markov Chain Monte Carlo (MCMC) algorithm was introduced for model selection purposes. This was applied to the problem of finding the joint Maximum *a posteriori* estimate for the number of states, model parameters and segmentation into these states. Simulated annealing was used to optimise the model with the reversible jump sampler enabling movement between model spaces. Unsupervised segmentation could therefore be achieved in a single iterative process. This strategy proved extremely computationally intensive and it became apparent that the optimisation of more complicated models, for example those using large neighbourhood GMRF's to model coarse textures, would be beyond the capability of these algorithms.

Thus, in chapter 5 a new approach was developed. The Gibbs sampler, used previously to draw from the regularising Potts model was replaced by the Partial Decoupling algorithm. Partial Decoupling works by breaking the image lattice down into clusters of pixels, realised on a finer scale than the image regions, these are then sampled as complete entities. Such an algorithm exhibits improved convergence over both Gibbs sampling and its closest relative, the more well known Swendsen-Wang algorithm, when drawing from a hierarchical image model posterior distribution. Partial Decoupling is also an auxiliary variable tech-

nique and allows the specification of hyper-parameters on these variables which, if chosen carefully can speed up convergence still further. This provides the mechanism by which the features of chapter 3 might speed up the optimisation process. Such an avenue has been extensively explored in section 5.5 for the case of using Gabor filter functions to improve the convergence for the hierarchical GMRF model.

A further advantage to using such an algorithm is in the generation of proposals within the reversible jump sampling algorithm. Previously in the literature and in chapter 4 fresh model parameters had been generated randomly, a technique too inefficient to use with complicated models having large numbers of model parameters. The ability to break the image down into small clusters, each of which may be considered homogeneous through the generation of hyper-parameters from the observed data, allows the new model parameters required in the reversible jump sampler to be estimated directly from the data. This improved acceptance rates and thus convergence significantly. Results presented for Brodatz texture mosaics and a remotely sensed image demonstrate these advances. Furthermore, these algorithms allow unsupervised segmentation to proceed with the incorporation of line process into the image model.

The work described in this dissertation provides mechanisms which could be used in many further applications and studies. For example, an extensive study assessing the effectiveness of the traditional image models would appear highly pertinent at this date. For crop discrimination the models used throughout this dissertation are unlikely to be sufficient since observed gray-level often drifts across large open areas in remotely sensed images. Development of suitable alternatives would be a major advance, made possible through the use of algorithms of the type described here. The consideration of more complex line processes, with the possibility of linking such an algorithm to higher level models and optimisation processes, for example Arak processes [22], would be of much interest. Finally, the algorithms of chapter 5 could be applied to the optimisation of Bilateral Markov Models for time series, an area of research as yet not fully investigated.

Mean Field Derivation

A

There follows a re-working of the derivation by Geiger & Girosi[34] of the mean estimates of the weak membrane model line elements. The notation is given in section 2.5.3.

Consider the expression for the mean value for a single horizontal field element h_{ij},

$$\begin{aligned}\langle h_{i,j}\rangle &= \frac{1}{Z}\sum_{\mathbf{x}}\exp\Big\{-\beta\sum_{i,j}\Big[\frac{(y_{i,j}-x_{i,j})^2}{2\sigma^2}+\gamma^h_{i,j}+\gamma^v_{i,j}\Big]\Big\} \\ &\quad\times\sum_{\mathbf{h},\mathbf{v}}h_{i,j}\exp\Big\{-\beta\sum_{i,j}\big[(1-h_{i,j})G^h_i,j+(1-v_{i,j})G^v_{\ i,j}\big]\Big\}\end{aligned} \tag{A.1}$$

where, $G^h_{i,j}=(x_{i,j}-x_{i+1,j})^2-\gamma^h_{i,j}$ and $G^v_{\ i,j}=(x_{i,j}-x_{i,j+1})^2-\gamma^v_{i,j}$. Because all $h_{i,j}$ and $v_{i,j}$ are defined on the real line segment $[0,1]$, the above equation may be re-written

$$\langle h_{i,j}\rangle=-\frac{1}{\beta Z}\sum_{\mathbf{x}}\exp\Big\{-\beta\sum_{i,j}\Big[\frac{(y_{i,j}-x_{i,j})^2}{2\sigma^2}+\gamma^h_{i,j}+\gamma^v_{i,j}\Big]\Big\}\frac{\partial Z_{hv}(\mathbf{x})}{\partial\gamma^h_{i,j}} \tag{A.2}$$

where $Z_{h,v}(\mathbf{x})=\sum_{\mathbf{h},\mathbf{v}}\exp\{-\beta\sum_{i,j}[h_{i,j}G^h_i,j+v_{i,j}G^v_{\ i,j}]\}$. If the free energy of the system $Z_{h,v}(\mathbf{x})$ is defined so that

$$Z_{h,v}(\mathbf{x})=e^{-\beta F_{hv}(\mathbf{x})} \tag{A.3}$$

then equation A.2 may be re-expressed,

$$\langle h_{i,j}\rangle=-\frac{1}{\beta Z}\sum_{\mathbf{x}}\exp\Big\{-\beta\sum_{i,j}\Big[\frac{(y_{i,j}-x_{i,j})^2}{2\sigma^2}+\gamma^h_{i,j}+\gamma^v_{i,j}\Big]+F_{hv}(\mathbf{x})\Big\}\frac{\partial F_{hv}(\mathbf{x})}{\partial\gamma^h_{i,j}} \tag{A.4}$$

Thus, the mean of each line element may be calculated by finding the statistical mean of $\frac{\partial F_{hv}(\mathbf{x})}{\partial\gamma^h_{i,j}}$ over the field $\mathbf{x}$, i.e.,

$$h_{i,j} = \Big\langle\frac{\partial F_{hv}(\mathbf{x})}{\partial\gamma^h_{i,j}}\Big\rangle_{\mathbf{x}} \tag{A.5}$$

If the line elements are assumed statistically independent, then $Z_{h,v}(\mathbf{x})$ will consist of a non-interactive spin system in an external field. Hence, the summation over both

horizontal and vertical fields may calculated analytically so that

$$Z_{h,v}(\mathbf{x}) = \prod_{i,j} \left(1 + e^{-\beta G^h_{i,j}}\right)\left(1 + e^{-\beta G^h_{i,j}}\right) \tag{A.6}$$

Using this expression to find the free energy $F_{hv}(\mathbf{x})$, gives the expression for the statistical mean of a horizontal line element:

$$h_{i,j} \;=\; \left\langle \frac{1}{1 + e^{-\beta G^h_{i,j}}} \right\rangle_{\mathbf{x}} \tag{A.7}$$

A similar expression may be obtained for the vertical line process.

Reversible Jumps

B

There follows a proof of detailed balance for the reversible jump sampler [43] as applied by Richardson & Green[82] to the problem of MAP estimation for an unknown number of components in a Gaussian mixture model, a form of the hidden data problem. The first section of the proof is similar to that presented by Green[43] when deriving the reversible jump acceptance ratio. The derivation showing that such an acceptance ratio satisfies detailed balance is original.

If a Markov Chain enjoys detailed balance with respect to a distribution $\pi(\mathbf{x})$, then $\pi(\mathbf{x})p(\mathbf{x},\mathbf{x}') = \pi(\mathbf{x}')p(\mathbf{x}',\mathbf{x})$. The Metropolis-Hastings algorithm guarantees this equality by incorporating an acceptance ratio $\alpha(\mathbf{x},\mathbf{x}')$ into the transition probability such that $p(\mathbf{x},\mathbf{x}') = q(\mathbf{x},\mathbf{x}') \times \alpha(\mathbf{x},\mathbf{x}')$, where $q(\mathbf{x},\mathbf{x}')$ is the proposal distribution. To find a value for the acceptance probability the incorporation of the new transition probability into the expression for detailed balance must be considered. To facilitate this an expression is needed for the transition kernel of such an accept/reject sampler. If $x \in C$, where C is the complete parameter space, then the transition kernel for a move to $x' \in B$, where $B \subset C$ and is a Borel set, is given by

$$p(x,B) \;=\; \int_B q(x,dx')\;\alpha(x,x') \;+\; s(x)\;\mathrm{I}[x \in B] \tag{B.1}$$

where $I[\cdot]$ is the indicator function and $s(x)$ the probability of staying at x through either a rejection or no move being proposed, is defined

$$s(x) \;=\; \int_C q(x,dx')\;[1-\alpha(x,x')] \;+\; 1 \;-\; q(x,C) \tag{B.2}$$

For detailed balance to be preserved when considering the moves from the set A to B, or the reverse, from B to A,

$$\begin{aligned}\int_A \pi(dx) \int_B q(x,dx')\;\alpha(x,x') \;+\; \int_{A\cap B} \pi(dx)\;s(x) \\ =\; \int_B \pi(dx') \int_A q(x',dx)\;\alpha(x',x) \;+\; \int_{A\cap B} \pi(dx')\;s(x')\end{aligned} \tag{B.3}$$

A sufficient condition for this equality and hence detailed balance to be ensured is

$$\int_A \pi(dx) \int_B q(x, dx')\ \alpha(x, x') \ = \ \int_B \pi(dx') \int_A q(x', dx)\ \alpha(x', x) \tag{B.4}$$

If $\pi(dx)\ q(x, dx')$ has a finite density $f(x, x')$ with respect to a symmetric measure ξ on $C \times C$, then if $\alpha(x, x')\ f(x, x') \ = \ \alpha(x', x)\ f(x', x)$, then

$$\begin{aligned} \int_A \pi(dx) \int_B q(x, dx')\ \alpha(x, x') \quad &= \quad \int_A \int_B \xi(dx, dx')\ f(x, x')\ \alpha(x, x') \\ &= \quad \int_A \int_B \xi(dx', dx)\ f(x', x)\ \alpha(x', x) \\ &= \quad \int_B \pi(dx') \int_A q(x', dx)\ \alpha(x', x) \end{aligned} \tag{B.5}$$

and detailed balance is maintained. To ensure the above condition, the acceptance ratio is set to

$$\alpha(x, x') \ = \ \min\left[1, \frac{f(x', x)}{f(x, x')}\right] \ = \ \min\left[1, \frac{\pi(dx')q(x', dx)}{\pi(dx)q(x, dx')}\right] \tag{B.6}$$

To apply this proof to the reversible jump problem where parameter spaces differ in dimension before and after the proposed move, requires some re-working. Consider moving from a k-dimensional model parameterised by $\boldsymbol{\theta}^{(k)}$ to a k'-dimensional model parameterised by $\boldsymbol{\theta}^{(k')}$. To ensure a corresponding reverse move is possible for each forward proposal, there must be a bijection between parameter spaces. Because these differ in dimension it is necessary to extend the parameter vectors by random variable vectors $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(k')}$ so that $\#[\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)}] = \#[\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')}]$, where $\#[\cdot]$ indicates the operator, 'the dimension of'. A bi-directional transformation between the extended parameter spaces is now possible, $[\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)}] \leftrightarrow [\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')}]$. This interrelationship between the extended vectors is defined by a pair of invertible functions $[\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')}] = \theta'(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})$ and $[\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)}] = \theta(\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')})$, i.e. $\theta_k(\cdot)$ is the inverse function for $\theta_{k+1}(\cdot)$.

Returning to equation B.4, the sufficient condition for detailed balance may be re-written

$$\begin{aligned} &\int_{\boldsymbol{\theta}^{(k)}} \pi(d\boldsymbol{\theta}^{(k)}) \int_{\mathbf{u}^{(k)}} q\big(\boldsymbol{\theta}^{(k)},\ \theta'(\boldsymbol{\theta}^{(k)}, d\mathbf{u}^{(k)})\big)\ \alpha(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^{(k')}) \\ &\qquad = \ \int_{\boldsymbol{\theta}^{(k')}} \pi(d\boldsymbol{\theta}^{(k')}) \int_{\mathbf{u}^{(k')}} q\big(\boldsymbol{\theta}^{(k')},\ \theta(\boldsymbol{\theta}^{(k')}, d\mathbf{u}^{(k')})\big)\ \alpha(\boldsymbol{\theta}^{(k')}, \boldsymbol{\theta}^{(k)}) \end{aligned} \tag{B.7}$$

Consider the right hand side of the above equation: by Fubini's theorem[13] and the

change of variable theorem for a Lesbegue Integral (see Billingsley[13], theorem 17.2),

$$\begin{aligned}
&\int_{\boldsymbol{\theta}^{(k')}} \pi(d\boldsymbol{\theta}^{(k')}) \int_{\mathbf{u}^{(k')}} q\big(\boldsymbol{\theta}^{(k')},\ \theta(\boldsymbol{\theta}^{(k')}, d\mathbf{u}^{(k')})\big)\ \alpha(\boldsymbol{\theta}^{(k')}, \boldsymbol{\theta}^{(k)}) \qquad \text{(B.8)}\\
&\quad = \int_{\boldsymbol{\theta}^{(k')}\times\mathbf{u}^{(k')}} \pi(\boldsymbol{\theta}^{(k')})\ q\big(\boldsymbol{\theta}^{(k')},\ \theta(\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')})\big)\ \lambda\big(\partial(\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')})\big)\ \alpha(\boldsymbol{\theta}^{(k')}, \boldsymbol{\theta}^{(k)})\\
&\quad = \int_{\boldsymbol{\theta}^{(k)}\times\mathbf{u}^{(k)}} \pi\big(\theta'(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)\ q\big(\theta'(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)}), [\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)}]\big)\\
&\qquad\qquad \times \left|\frac{\lambda\big(\partial(\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')})\big)}{\lambda\big(\partial(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)}\right| \lambda\big(\partial(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)\ \alpha(\boldsymbol{\theta}^{(k')}, \boldsymbol{\theta}^{(k)})
\end{aligned}$$

The condition for detailed balance (equation B.7) may now be expressed over a single measure space:

$$\begin{aligned}
&\int_{\boldsymbol{\theta}^{(k)}\times\mathbf{u}^{(k)}} \pi(\boldsymbol{\theta}^{(k)})\ q\big(\boldsymbol{\theta}^{(k)},\ \theta'(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)\ \lambda\big(\partial(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)\ \alpha(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^{(k')}) \qquad \text{(B.9)}\\
&\quad = \int_{\boldsymbol{\theta}^{(k)}\times\mathbf{u}^{(k)}} \pi\big(\theta'(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)\ q\big(\theta'(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)}), [\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)}]\big)\\
&\qquad\qquad \times \left|\frac{\lambda\big(\partial(\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')})\big)}{\lambda\big(\partial(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)}\right| \lambda\big(\partial(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)\ \alpha(\boldsymbol{\theta}^{(k')}, \boldsymbol{\theta}^{(k)})
\end{aligned}$$

It immediately follows that the forwards acceptance ratio required to fulfil this condition is given by

$$\alpha(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^{(k')}) = \min\left[1, \frac{\pi\big(\boldsymbol{\theta}^{(k')}\big)}{\pi\big(\boldsymbol{\theta}^{(k)}\big)} \frac{q\big(\boldsymbol{\theta}^{(k')}, \theta(\boldsymbol{\theta}^{(k')}, d\mathbf{u}^{(k')})\big)}{q\big(\boldsymbol{\theta}^{(k)}, \theta'(\boldsymbol{\theta}^{(k)}, d\mathbf{u}^{(k)})\big)} \left|\frac{\lambda\big(\partial(\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k')})\big)}{\lambda\big(\partial(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k)})\big)}\right|\right] \qquad \text{(B.10)}$$

Cluster Likelihood Derivation

C

To arrive at an expression for the joint distribution for the state variables comprising a single cluster given those over the remainder of the image it is necessary to split the state variables of the entire image into two groups, those contained in the cluster designated $\mathbf{n}$ and those external to the cluster, denoted $\mathbf{b}$.

The conditional distribution for the cluster given the remainder of the image is, by definition, $p(\mathbf{n} \mid \mathbf{b}) = \frac{p(\mathbf{n},\mathbf{b})}{p(\mathbf{b})}$. The conditional distribution may be found via this equation by first writing an expression for the numerator (comprising likelihood for the entire image) and then integrating out all contributions of the cluster, thus giving the denominator $p(\mathbf{b})$.

If the inverse covariance matrix is divided into non-equal quadrants corresponding to: within cluster correlations; correlations between the cluster pixels and finally correlations between the remaining pixels, i.e.

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{nn} & \mathbf{R}_{nb} \\ \mathbf{R}_{bn} & \mathbf{R}_{bb} \end{bmatrix} \tag{C.1}$$

then since, $\mathbf{R}_{nb} = \mathbf{R}_{bn}^T$, the joint likelihood for the complete state variable matrix may be written

$$p(\mathbf{n}, \mathbf{b}) \propto |\mathbf{R}|^{\frac{1}{2}} \exp\Big\{ -\frac{1}{2}\big[\mathbf{n}^T\mathbf{R}_{nn}\mathbf{n} + 2\mathbf{n}^T\mathbf{R}_{nb}\mathbf{b} + \mathbf{b}^T\mathbf{R}_{bb}\mathbf{b}\big]\Big\} \tag{C.2}$$

To find $p(\mathbf{b})$ requires the integration of the above joint distribution over all $\mathbf{n}$ which may be found by completing the Gaussian distribution:

$$\begin{aligned} p(\mathbf{b}) &\propto |\mathbf{R}|^{\frac{1}{2}} \int \exp\Big\{ -\frac{1}{2}\big[\mathbf{n}^T\mathbf{R}_{nn}\mathbf{n} + 2\mathbf{n}^T\mathbf{R}_{nb}\mathbf{b} + \mathbf{b}^T\mathbf{R}_{bb}\mathbf{b}\big]\Big\} d\mathbf{n} \\ &= |\mathbf{R}|^{\frac{1}{2}} \exp\Big\{ -\frac{1}{2}\mathbf{b}^T\mathbf{R}_{bb}\mathbf{b}\Big\} \int \exp\Big\{ -\frac{1}{2}\big[\mathbf{n}^T\mathbf{R}_{nn}\mathbf{n} + 2\mathbf{n}^T\mathbf{R}_{nb}\mathbf{b}\big]\Big\} d\mathbf{n} \\ &= \frac{|\mathbf{R}|^{\frac{1}{2}}}{|\mathbf{R}_{nn}|} \exp\Big\{ -\frac{1}{2}\big[\mathbf{b}^T\mathbf{R}_{bb}\mathbf{b} - \big[\mathbf{R}_{nn}^{-1}\mathbf{R}_{nb}\mathbf{b}\big]^T\mathbf{R}_{nb}\mathbf{b}\big]\Big\} \end{aligned}$$

Hence, the conditional distribution may be found:

$$\begin{aligned} p(\mathbf{n} \mid \mathbf{b}) &= \frac{p(\mathbf{n}, \mathbf{b})}{p(\mathbf{b})} \\ &\propto |\mathbf{R}_{nn}| \ \exp\Big\{ -\frac{1}{2}\big[\mathbf{n}^T\mathbf{R}_{nn}\mathbf{n} \ + \ 2\mathbf{n}^T\mathbf{R}_{nb}\mathbf{b} \ + \ \big[\mathbf{R}_{nn}^{-1}\mathbf{R}_{nb}\mathbf{b}\big]^T\mathbf{R}_{nb}\mathbf{b}\big]\Big\} \end{aligned}$$

Bibliography

[1] H. Akaike. Stochastic Theory of Minimal Realisation. *IEEE Trans. on Automatic Control*, AC-19(6):667–674, Dec 1974.

[2] D.A. Bader, J. Jájá, and R. Chellappa. Scalable Data Parallel Alogithms for Texture Synthesis using Gibbs Random Fields. *IEEE Trans. on Image Processing*, 4(10), Oct 1995.

[3] S.A. Barker, A.C. Kokaram, and P.J.W. Rayner. Unsupervised Segmentation of Images. *Proceedings of SPIE*, 3459, 1998.

[4] S.A. Barker and P.J.W. Rayner. Unsupervised Image Segmentation using Markov Random Field Models. *To appear in Pattern Recognition.*

[5] S.A. Barker and P.J.W. Rayner. Unsupervised Image Segmentation using Markov Random Field Models. *Lecture Notes in Computer Science*, 1997.

[6] S.A. Barker and P.J.W. Rayner. Unsupervised Image Segmentation. *Proceedings International Conference on Accoustics Speech and Signal Processing*, 1998.

[7] R.J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press Inc. (London) Ltd., 1982.

[8] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory.* Wiley, 1994.

[9] J. Besag. Spatial interaction and the statistical analysis of lattice. *J. Royal Statist. Soc., Series B*, 36:192–236, 1974.

[10] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

[11] J. Besag. On the statistical analysis of dirty pictures. *J. Royal Statist. Soc., Series B*, 48:259–302, 1986.

[12] J. Besag and P.J. Green. Spatial statistics and Bayesian computation. *J. Royal Statist. Soc., Series B*, 55:25–37, 1993.

[13] P. Billingsley. *Measure Theory.* Addison Wesley, 1960.

[14] C. Bouman and D. Liu. Multiple Resolution Segmentation of Textured Images. *IEEE Trans. Patt. Anal. Machine Intell.*, 13(2):99–113, Feb 1991.

[15] A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel Texture Analysis using Localized Spatial Filters. *IEEE Trans. Patt. Anal. Machine Intell.*, 12(1):55–73, Jan 1990.

[16] P. Brodatz. *Texture: A Photographic Album for Artists and Designers.* New York: Dover, 1966.

[17] F.W. Campbell and J.G. Robson. Application of Fourier analysis to visibility of gratings. *J.Physiol*, 197:551–566, 1968.

[18] B. Chalmond. An iterative gibbsian technique for reconstruction of m-ary images. *Pattern Recognition*, 22(6):747–761, 1989.

[19] D. Chandler. *Introduction to Modern Statistical Mechanics.* Oxford University Press, 1987.

[20] S. Chatterjee. Classification of Textures using Gaussian Markov Random Fields. In R. Chellappa and A.K. Jain, editors, *Markov Random Fields, Theory and Applications.* Academic Press Inc., 1993.

[21] R. Chellappa and S. Chatterjee. Classification of Textures using Gaussian Markov Random Fields. *IEEE Trans. Acoustics, Speech and Signal Processing*, 33(4):959–963, Aug 1985.

[22] P. Clifford and G. Nicholls. A Metropolis Sampler for Polygonal Image Reconstruction. *Department of Statistics, Oxford University*, Dec 1994.

[23] F.S. Cohen and D.B. Cooper. Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields. *IEEE Trans. Patt. Anal. Machine Intell.*, 9(2):195–219, Mar 1987.

[24] F.S. Cohen and Z. Fan. Maximum Likelihood unsupervised texture segmentation. *CVGIP:Graphical Models and Image Processing*, 54(3):239–251, May 1992.

[25] F.S. Cohen, Z. Fan, and S. Attali. Automated Inspection of Textile Fabrics Using Textural Models. *IEEE Trans. Pattern Analaysis and Machine Intelligence*, 13(8):803–808, 1991.

[26] M.L. Comer and E.J. Delp. Multiresolution Image Segmentation. *Proceedings*, pages 2415–2418, 1995.

[27] J. Daugman. Two-Dimensional Spectral Analysis of Cortical Receptive Field Profiles. *Vision Research*, 20:847–856, 1980.

[28] J. Daugman. Uncertainty relation for resolution in space, spatial freqency, and orientation optimized by two-dimensional visual cortical filters. *J.Opt.Soc.Am. A*, 2:1160–1169, jul 1985.

[29] J. Daugman. Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression. *IEEE Trans. Acoustics, Speech and Signal Processing*, 36(7):1169–1179, jul 1988.

[30] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal Royal Statistical Society, Series B.*, 1:1–38, 1977.

[31] J.C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters. *J. Cybernetics*, 3:32–57, 1974.

[32] R.G. Edwards and A.D. Sokal. Generalization of the Fortuin-Kasteleyn-swendsen-Wang representation and Monte-Carlo algorithm. *Physical Review D. Particles and Fields*, 38:2009–2012, 1988.

[33] D. Gabor. Theory of Communication. *J. Inst. Electr. Eng.*, 93:429–457, 1946.

[34] D. Geiger and F. Girosi. Parallel and deterministic algorithms for mrfs: surface reconstruction and integration. *IEEE Trans. Patt. Anal. & Machine Intell.*, 12(5):401–412, May 1991.

[35] D. Geiger and A. Yuille. A Common Framework for Image Segmentation. *International Journal of Computer Vision*, 6(3):227–243, 1991.

[36] D. Geman. Bayesian Image Analysis by Adaptive Annealing. *Proc. of International Geoscience and Remote Sensing Symposium, Amherst, USA*, pages 269–276, 1985.

[37] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary Detection by Constrained Optimization. *IEEE Trans. Patt. Anal. & Machine Intell.*, 12(7):609–628, July 1990.

[38] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(3):367–383, Mar 1992.

[39] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Trans. Patt. Anal. & Machine Intell.*, 6(6):721–741, Nov 1984.

[40] B. Gidas. A Multilevel-Multiresolution Technique for Computer Vision via Renormalization Group Ideas. *Proc. SPIE, High Speed Computing*, 880:214–218, 1988.

[41] B. Gidas. A Renormalization Group Approach to Image Processing Problems. *IEEE Trans. Patt. Anal. Machine Intell.*, 11(2):164–180, Feb 1989.

[42] A.J. Gray. Discussion on Gibbs sampler and other MCMC methods. *J. Royal Statist. Soc., Series B*, 55:58–61, 1993.

[43] P.J. Green. Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1996.

[44] F. Gustafsson and H. Hjalmarsson. Twenty-one Estimators for Model Selection. *Automatica*, 31(10):1377–1392, 1995.

[45] R.M. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[46] R.M. Haralick, K. Shanmugan, and I. Dinstein. Texture Features for Image Classification. *IEEE Trans. Syst. Man. Cybernet.*, 3(6):610–621, 1973.

[47] T.J. Herbert and K. Lu. Expectation-Maximization Algorithms, Null Spaces and MAP Image Restoration. *IEEE Trans. Image Processing*, 4(8):1084–1095, Aug 1995.

[48] B.L. Hickman, M.P. Bishop, and M.V. Rescigno. Advance Computational Methods for Spatial Information Extraction. *Computers and Geosciences*, 21(1):153–173, Feb 1995.

[49] D. Higdon. Discussion on Gibbs sampler and other MCMC methods. *J. Royal Statist. Soc., Series B*, 55:78, 1993.

[50] D. Higdon. Auxiliary Variable Methods for Markov Chain Monte Carlo with Applications. *unpublished preprint - Duke University*, Jul 1996.

[51] M. Hurn and C. Jennison. An Extension to Geman and Reynolds' Approach to Constrained Restoration and the Recovery of Discontinuities. *IEEE Trans. Patt. Anal. Machine Intell.*, 18(6):657–662, Jun 1996.

[52] A.K. Jain and F. Farrokhnia. Unsupervised Tecture Segmentation using Gabor Filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[53] F.C. Jeng, J.W. Woods, and S. Rastogi. Compound Gauss-Markov Random Fields fo Parallel Image Processing. In R. Chellappa and A.K. Jain, editors, *Markov Random Fields, Theory and Applications*. Academic Press Inc., 1993.

[54] R. Chellappa (Editors) L.M. Kanal and A. Rosenfeld. Two-Dimensional Discrete Gaussian Markov Random Field Models for Image Processing. *Progress in Pattern Recognition 2, Elsevier Science Publishers B.V. (North-Holland)*, pages 79–112, 1985.

[55] R.L. Kashyap. Inconsistency of the AIC Rule for Estimating the Order of Autoregressive Models. *IEEE Trans. Automat. Control*, pages 996–998, 1980.

[56] R.L. Kashyap. Analysis and Synthesis of Image Patterns by Spacial Interaction Models. In L.N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*. Amsterdam. The Netherlands: North-Holland, 1981.

[57] R.L. Kashyap. Finite Lattice Random Field Models for Finite Images. *Proc. of Annual Conf. Inf. Sci. Syst.*, pages 215–220, 1981.

[58] R.L. Kashyap. Optimal Choice of AR and MA parts in Autoregressive Moving Average Model. *IEEE Trans. Patt. Anal. and Mach. Int.*, pages 99–104, 1982.

[59] Z. Kato, J. Zerubia, and M. Berthod. Unsupervised Adaptive Image Segmentation. *Proc. ICASSP95*, pages 2399–2402, 1995.

[60] Z. Kato, J. Zerubia, and M. Berthod. Unsupervised Parallel Image Classification using a Hierarchical Markovian Model. *Proc. ICCV95*, pages 169–174, 1995.

[61] C. Kervrann and F. Heitz. A Markov Random Field model-based approach to unsupervised texture segmentation using local and global statistics. *IEEE Trans. Image Processing*, 4(6):856–862, June 1995.

[62] S. Krishnamachari and R. Chellappa. Multiresolution GMRF Models for Texture Segmentation. *Proceedings ICASSP*, pages 2407–2410, 1996.

[63] S. Krishnamachari and R. Chellappa. Multiresolution Gauss-Markov Random Field Models for Texture Segmentation. *IEEE Transactions on Image Processing*, 6(2):251–267, Feb 1997.

[64] S. Lakshmanan and H. Derin. Simultaneous Parameter Estimation and Segmentation of Gaussian Random Fields using Simulated Annealing. *IEEE Trans. Patt. Anal. & Machine Intell.*, 11:799–813, Aug 1989.

[65] S. Lakshmanan and H. Derin. Gaussian Markov Random Fields at Multiple Resolutions. In R. Chellappa and A.K. Jain, editors, *Markov Random Fields, Theory and Applications*. Academic Press Inc., 1993.

[66] D.A. Langan, J.W. Modestino, and J. Zhang. Cluster Validation for Unsupervised Stochastic model-based Image Segmentation. *Proc. ICIP94*, pages 197–201, 1994.

[67] S.Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag Tokyo, 1995.

[68] B.S. Manjunath and R. Chellappa. Unsupervised texture segmentation using Markov Random Fields. *IEEE Trans. Patt. Anal. & Machine Intell.*, 13(5):478–482, May 1991.

[69] B.S. Manjunath and W.Y. Ma. Texture Features for Browsing and Retreival of Image Data. *IEEE Trans. Patt. Anal. & Machine Intell.*, 18(8):837–842, Aug 1996.

[70] B.S. Manjunath, T. Simchony, and R. Chellappa. Stochastic and Deterministic Networks for Texture Segmentation. *IEEE Trans. Acoustics, Speech and Signal Proc.*, 38(6):1039–1049, Jun 1990.

[71] S. Marcelja. Mathematical description of the responses of simple cortical cells. *J.Opt.Soc.Am*, 70(11):1297–1300, Nov 1980.

[72] E. Marinari and G. Parisi. Simulated Tempering: a new Monte Carlo scheme. *Europhysics Letters*, 19(6):451–458, Jul 1992.

[73] R.D. Morris. *Image Sequence Restoration using Gibbs Distributions.* PhD thesis, Univeristy of Cambridge, U.K., May 1995.

[74] R. Muzzolini, Y.H. Yang, and R. Pierson. Texture Characterization using Robust Statistics. *Pattern Recognition*, 27(1):119–134, 1994.

[75] R. Neal. Probabilistic Inference using Markov Chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, Univeristy of Toronto, Canada, 1993.

[76] H.H. Nguyen and P. Cohen. Gibbs Random Fields, Fuzzy Clustering, and the unsupervised segmentation of images. *CVGIP:Graphical Models and Image Processing*, 55(1):1–19, Jan 1993.

[77] J.J.K. Ó Ruanaidh and W.J. Fitzgerald. *Numerical Bayesian Methods Applied to Signal Processing.* Springer-Verlag, New York, 1996.

[78] D.K. Panjwani and G. Healey. Markov Random Field models for unsupervised segmentation of textured color images. *IEEE Trans. Patt. Anal. & Machine Intell.*, 17(10):939–954, Oct 1995.

[79] A. Papoulis. *Systems and Transformations with Applications in Optics.* McGraw-Hill, New York, 1968.

[80] D.A. Pollen and S.F. Ronner. Phase relationships between adjacent simple cells in the visual cortex. *Science*, 212:1409–1411, 1981.

[81] M. Porat and Y.Y. Zeevi. The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision. *IEEE Trans. Patt. Anal. & Machine Intell.*, 10(4):452–467, Jul 1998.

[82] S. Richardson and P.J. Green. On Bayesian analysis of mixtures with an unknown number of components. *J.Roy.Stat.Soc.Series B*, 59(4):731–792, 1997.

[83] J. Rissanen. Modeling by Shortest Data Description. *Automatica*, 14:465–471, 1978.

[84] Y. Sakamoto, M. Ishiguro, and G. Kitagawa. *Akaike Information Criterion Statistics*. KTK Scientific Publishers/Tokyo, D.Reidel Publishing Company, Kluwer Academic Publishers Group, 1986.

[85] G. Schwartz. Estimating the Dimension of a Model. *Annals of Statistics*, 6:461–464, 1978.

[86] G. Sharma and R. Chellappa. Two-Dimensional Maximum Entropy Power Spectra. *IEEE Transactions on Information Theory*, 35:90–99, Jan 1985.

[87] J.V. Soares, C.D. Renno, A.R. Formaggio, C.C.F. Yanasse, and A.C. Frery. An Investigation of the Selection of Texture Features for Crop Discrimination Using SAR Imagery. *Remote Sensing of Environment*, 59(2):234–247, 1997.

[88] R. Srichander. Efficient Schedules for Simulated Annealing. *Engineering Optimization*, 24:161–176, 1995.

[89] P.N. Strenski and S. Kirkpatrick. Analysis of Finite Length Annealing Schedules. *Algorithmica*, 6:346–366, 1991.

[90] R.H. Swendsen and J. Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88, Jan 1987.

[91] H. Szu and R. Hartley. Fast Simulated Annealing. *Physics Letters A.*, 122:157–162, 1987.

[92] M.A. Tanner. *Tools for Statistical Inference*. Springer-Verlag, 1993.

[93] L. Tierny. Markov Chains for exploring posterior distributions. *Annals of Statistics*, 22(5):1701–1762, 1994.

[94] M.R. Turner. Texture Discrimination by Gabor Functions. *Bilogical Cybernetics*, 55:71–82, 1986.

[95] R.D. De Valois, D.G. Albrecht, and L.G. Thorell. Spatial-frequency selectivity of cells in macaque visual cortex. *Vision Res.*, 22:545–559, 1982.

[96] C.S. Won and H. Derin. Unsupervised Segmentation of Noisy and Textured Images using Markov Random Fields. *CVGIP:Graphical Models and Image Processing*, 54(4):308–328, July 1992.

[97] J.W. Woods. Two-Dimensional Discrete Markovian Fields. *IEEE Transactions on Information Theory*, 18(2):232–240, Mar 1972.

[98] M.D. Wu. *Markov Chain Monte Carlo Methods Applied to Bayesian Data Analysis.* PhD thesis, University of Cambridge, U.K., Nov 1997.

[99] J. Zerubia and R. Chellappa. Mean Field Annealing Using Compound Gauss-Markov Random Fields for Edge Detection and Image Estimation. *IEEE Transactions on Neural Networks*, 4(4):703–709, Jul 1993.

[100] J. Zhang. The Mean Field Theory in EM Procedures for Markov Random Fields. *IEEE Transactions on Signal Processing*, 40(10):2570–2583, Oct 1992.

[101] J. Zhang and J.W. Modestino. A Model Fitting Approach to Cluster Validation with application to Stochastic Model-Based Image Segmentation. *Proceedings ICASSP*, 2:1148–1151, 1998.

[102] J. Zhang, J.W. Modestino, and D.A. Langan. Maximum-Likelihood Parameter Estimation for Unsupervised Stochastic Model-Based Image Segmentation. *IEEE Transactions on Image Processing*, 3(4):404–420, July 1994.