

University of Glasgow
Faculty of Information and Mathematical Sciences
Department of Statistics

Master Thesis

Adaptive Bayesian Sampling with Application to ‘Bubbles’



UNIVERSITY
of
GLASGOW

submitted by Ekaterina Ignatieva (0604405)
born on 04.05.1983 in St.-Petersburg

Supervisor: **Prof. Dr. Nial Friel**

Glasgow, 11th August 2008

CONTENTS

1. <i>Introduction</i>	15
2. <i>Placing Bubbles in a Bayesian Setting</i>	20
2.1 Original Bubbles with Adaption	23
2.2 Drawbacks and Further Improvement	24
3. <i>Adaptive Bubbles with Beta MRF Prior</i>	26
3.1 Markov Random Fields and MCMC	27
3.1.1 The Metropolis-Hasting Algorithm	28
3.1.2 The Gibbs Sampler	29
3.2 Adaptive Bubbles with Beta MRF Prior: the Procedure	30
3.3 Modeling the joint distribution.	31
3.3.1 For one image pixel	31

3.3.2	For the whole image	35
3.4	Modeling the Likelihood.	39
3.5	Modeling posterior distribution.	40
3.6	The Metropolis-Hasting Algorithm	41
3.7	Simulated Experiments.	42
3.7.1	Replicating Bubbles situation: Data generation and simulation behavior of the observer	43
3.7.2	Tuning Parameters	45
4.	<i>Adaptive Bubbles with Ising Prior</i>	52
4.1	Ising and Autologistic Models	53
4.2	Adaptive Bubbles with Ising prior: the Procedure	57
4.3	Simulated Experiments	60
4.3.1	Tuning Parameters	61
5.	<i>Applications to Real Data Examples</i>	73
5.1	Experiment 1: EXNEX	74
5.2	Experiment 2: GENDER	85

<i>6. Inference for hyperparameters</i>	<i>94</i>
6.1 <i>Parameter Inference in Beta MRFs</i>	<i>96</i>
6.2 <i>Simulated Annealing Algorithm</i>	<i>97</i>
<i>7. Summary and Further Research</i>	<i>101</i>

LIST OF FIGURES

1.1	Faces used for judgments of gender and a face expression. . . .	15
1.2	Partial information used for judgments of gender.	16
1.3	Stimulus generation process. Upper panel shows the face decomposed into five independent scales. Middle panel: bubbles sample the information space at random locations, allowing overlapping. Bottom panel shows how bubbles are applied to the appropriate scales to produce a sub-sample of the face information.	17
3.1	Some realizations of Beta Markov Random Field after 1000 MCMC updates obtained by running the Metropolis algorithm with Gibbs update and parameters $\alpha = 3.0, \beta = 3.0, \theta = 3.0$ (upper left), $\alpha = 2.0, \beta = 2.0, \theta = 2.5$ (upper right), $\alpha = 1.5, \beta = 1.5, \theta = 2.5$ (lower left) and $\alpha = 1.0, \beta = 1.0, \theta = 2.0$ (lower right).	34

-
- 3.2 Cumulative distribution function (c.d.f.) of Beta - distribution is used as a classification pattern: the x -axis represents the proportion of object pixels among all revealed and the values of the c.d.f. correspond to the probability of correct classification. 44
- 3.3 The true 50×50 image. Proportion of object pixels in the image $\approx 24\%$. We use c.d.f. of the Beta (1, 3) distribution to simulate the behavior of the observer. 48
- 3.4 Estimated posterior probability map after different number of updates. Estimated using Beta MRF with parameters $\alpha = 2$, $\beta = 2$, $\theta = 2.5$ 49
- 3.5 Error rates after different numbers of trials for the cross example: original Bubbles (blue line), original Bubbles with Bayesian adaptation (green line) and adaptive Bubbles with Beta MRF prior (red line). 49
- 3.6 The true 100×100 image. Proportion of object pixels in the image $\approx 38\%$. We use c.d.f. of Beta (1, 2) distribution to simulate behavior of the observer. 50

3.7	Estimated posterior probability map after different number of updates. Estimated using Beta MRF with parameters $\alpha = 2$, $\beta = 2$, $\theta = 2.5$	50
3.8	Error rates after different numbers of trials for the cameraman example: original Bubbles (blue line), original Bubbles with Bayesian adaption (green line) and adaptive Bubbles with Beta MRF prior (red line).	51
4.1	Some realizations of binary Markov Random Field after 1000 MCMC updates with $\beta = (0, 0.4)$ (upper left), $\beta = (0, 0.2)$ (upper right), $\beta = (0.02, 0.4)$ (lower left) and $\beta = (-0.02, 0.4)$ (lower right). Simulated using Ising model.	55
4.2	Estimated posterior expectation in original Bubbles after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels are revealed.	64
4.3	Estimated posterior expectation in adaptive Bubbles with Ising prior after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels is revealed. Number of MCMC iterations is 1000.	65

4.4	Error rates after different numbers of trials: original Bubbles vs. original Bubbles with Bayesian adaption (upper panel) and original Bubbles vs. adaptive Bubbles with Ising prior (lower panel). . . .	66
4.5	Estimated posterior expectation in the original Bubbles approach after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels are revealed.	67
4.6	Estimated posterior expectation in adaptive Bubbles with Ising prior after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels are revealed. Number of MCMC iterations is 1000.	68
4.7	Error rates after different numbers of trials: original Bubbles vs. original Bubbles with Bayesian adaption (upper panel) and original Bubbles vs. adaptive Bubbles with Ising prior (lower panel). . . .	69
4.8	Estimated posterior expectation in original Bubbles approach after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels is revealed.	70

4.9	Estimated posterior expectation in adaptive Bubbles with Ising prior after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels is revealed. Number of MCMC iterations is 1000.	71
4.10	Error rates after different numbers of trials: original Bubbles vs. original Bubbles with Bayesian adaption (upper panel) and original Bubbles vs. adaptive Bubbles with Ising prior (lower panel). . . .	72
5.1	Faces used in experiment 1.	74
5.2	Bubble mask in original Bubbles (left panel) and adaptive Bubbles with Beta MRF/Ising prior (right panel).	75
5.3	Estimated posterior probability map for the original Bubbles (first column), original Bubbles with adaption (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column) after different numbers of trials. The number of MCMC iterations run between each update (every 100 trials) for the adaptive Bubbles with Beta MRF/Ising prior is 1000.	77

-
- 5.4 Weighting function computed using posterior probability values after the first update (i.e. after running 1000 MCMC iterations for the first time). Image pixels will be sampled with this weights in the following trials. 78
- 5.5 Actual performance: proportion of times observer has been correct with a particular expression to the total number of times this expression has been presented. For 'neutral' (upper panel) and 'happy' (lower panel). 80
- 5.6 Estimated posterior probability map after performing z-scoring at 90%-level to the estimates in Figure 5.3 for the original Bubbles (first column), original Bubbles with Bayesian adaption (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column). 81
- 5.7 Pixels with posterior probability values of 0.95 and higher for the original Bubbles (first column), original Bubbles with Bayesian adaption (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column). 82

5.8	Optimal stopping rule: how many pixels with high posterior probability values have been revealed after different number of trials (upper panel) and the Euclidean distance (lower panel) after different number of trials.	84
5.9	Faces used in experiment 2.	85
5.10	Bubble mask in the original Bubbles (left panel) and the adaptive Bubbles with Beta MRF/Ising prior (right panel).	86
5.11	Estimated posterior probability map for the original Bubbles (first column), original Bubbles with Bayesian adaption (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column) after different numbers of trials. The number of MCMC iterations run between each update (every 100 trials) for the adaptive Bubbles with Beta MRF/Ising prior is 1000.	87
5.12	Actual performance: proportion of times observer has been correct with a particular expression to the total number of times this expression has been presented. For 'male' (upper panel) and 'female' (lower panel).	89

5.13 Estimated posterior probability map after performing z-scoring at 90%-level to the estimates in Figure 5.11 for the original Bubbles (first column), original Bubbles with adaption (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column). 90

5.14 Pixels with high posterior probability values for the original Bubbles (first column), original Bubbles with adaption (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column). 92

5.15 Optimal stopping rule: how many pixels with high posterior probability values have been revealed after different number of trials (upper panel) and the Euclidean distance (lower panel) after different number of trials. 93

LIST OF TABLES

4.1	Likelihood ratio and ratio of priors.	59
6.1	Pseudo-likelihood estimation for Beta MRFs parameters α, β and θ . Estimated using simulated annealing algorithm.	100

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor, Prof. Dr. Nial Friel for the excellent guidance and support through the whole time of my Master studies. Without the commitment and his energy, this work would not exist in its present form. I would like to express my deep gratitude to him for constant encouragement and numberless helpful ideas and suggestions.

Moreover, it is a great pleasure for me to thank Prof. Dr. Philippe G. Schyns and Dr. Marie L. Smith for having helped me running experiments, for providing me with the data and for helpful discussions offering valuable comments and assistance during the whole year of my research.

Finally, I want to thank my friends and colleagues at the Statistics Department of Glasgow University for providing support and offering a friendly atmosphere, and also Novartis who has funded my studies.

Glasgow, 11th August 2008

Katja Ignatieva

1. INTRODUCTION

In an everyday life, people deal with different characterization tasks such as characterization of objects, judgments of gender, identity, age or a facial expression. For example, if we would ask a human observer to characterize two images in Figure 1.1 according to a gender and a facial expression, he would probably have no problem to classify the first image as a woman, with a happy expression and a second image as a man, with a neutral expression. That is, the observer can make judgements about gender and facial expressions based on the same input information. Now, imagine that only partial image information is revealed to the observer, as in a left panel of Figure 1.2. Could he confidently determine a gender based on this input information?



Fig. 1.1: Faces used for judgments of gender and a face expression.



Fig. 1.2: Partial information used for judgments of gender.

Right panel Figure 1.2 reveals additional input information which should improve gender judgement (i.e. male). Therefore, correct characterization tends to require different visual information from the same input. However, there is no unique method that can isolate information used in a certain characterization task. The Bubbles technique (Gosselin & Schyns 2001), in the following referred to as the *original Bubbles*, was developed to decide, which part of input information is used by the observer for solving a certain categorization problem.

The experimental situation presented in the original Bubbles reveals facial image information by a certain number of trials using random sampling of image pixels, Figure 1.3. Each trial represents a number of randomly located Gaussian windows, so called 'bubbles', which displays only a small portion of the image. After every trial the observer facing an image has to classify the sampled information (based on this partial information) as e.g. male or female. These trials are then repeated many times to produce one *experi-*

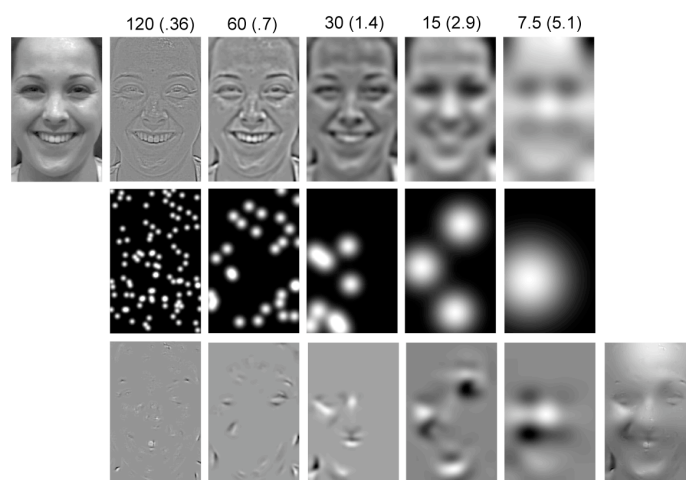


Fig. 1.3: Stimulus generation process. Upper panel shows the face decomposed into five independent scales. Middle panel: bubbles sample the information space at random locations, allowing overlapping. Bottom panel shows how bubbles are applied to the appropriate scales to produce a sub-sample of the face information.

mental stimulus, i.e. the ultimately collected data after all trials, see bottom right image in Figure 1.3. This stimulus is then used to infer which pixels are important for the classification. In visual cognition tasks such as judging the identity, gender or expression of a face, a human observer facing an image has to make a discrete classification based on this partial information.

The original Bubbles suggests that the regions for sampling (used to reveal facial image from) are chosen uniformly at random. In particular, the information gained after a certain number of trials is not used in determining

the information presented at the next trial. Exhaustive sampling leads to a number of trials equal to thousands per observer, which are very time-consuming and expensive to implement. In addition, the original Bubbles does not allow spatial information in the image to be modeled a priori. In this context, several questions addressed towards effective sampling of input information can be formulated: 'How can spatial dependence be incorporated in the image? How can the information gathered in previous samples be used in the decision making procedure? How can already classified image pixels be identified and removed from further sampling? Finally, how can the number of sampling trials be minimized?' Motivated by these questions, we aim to explore several statistical challenges below:

- Place Bubbles in a Bayesian setting.
- Objectively choose a stopping rule, which determines when the sampling ends and so, reduce the number of sampling trials.
- Incorporate spatial dependence in the image.

The thesis is organized as follows: the *Bayesian Bubbles* procedure which places the original Bubbles in a Bayesian setting is discussed in chapter 2. This chapter also introduces the *original Bubbles with adaption*, which addresses the problem of ineffective sampling in original Bubbles. Further im-

provement can be achieved by incorporating spatial dependence in the image using Markov Random Fields (MRFs): the *adaptive Bubbles with Beta MRF prior* and the *adaptive Bubbles with Ising prior* are developed in chapters 3 and 4 and applied to real data in chapter 5. Finally, chapter 6 discusses inference for hyperparameters and chapter 7 summarizes the results and provides an overview for future research.

2. PLACING BUBBLES IN A BAYESIAN SETTING

The original Bubbles (Gosselin & Schyns 2001) is not a Bayesian procedure. This chapter aims to put it in a Bayesian setting. We consider the Bubbles problem as a type of a latent variable problem. Imagine that we have a true *hidden* or *latent* image p with pixels labeled as 'important' or 'unimportant' for classification of a facial image as e.g. GENDER: male vs. female or EXNEX: expressive or not expressive. Our primary goal is to make posterior inference for the latent probability values p . In this chapter, the latent image p is assumed to have values in interval $[0, 1]$ which will be modeled using *Beta MRF*. Later (chapter 4), the latent points will take values in $\{-1, +1\}$ and thus, the *Ising MRF* will be applied to capture binary structure of latent image.

For a given pixel i , we denote by p_i the true proportion of correct choices or the true probability of correct classification. Let y denote the observed binary data (the outcome of any of the binary trials in Bubbles method), i.e. $y \in \{0, 1\}$: $y = 1$ for a correct response and $y = 0$ for an incorrect response.

In the following, we use the notation p_i^j to define the hidden probability value for pixel i after trial j ; p^j refers then to the whole lattice after trial j . Similarly, y_i^j and y^j define a single value for pixel i or the whole observed lattice after trial j , respectively. Then the *Likelihood function* for each pixel i after the first trial can be written in the following way:

$$L(y_i^1|p_i) = p_i^{y_i^1} (1 - p_i)^{1-y_i^1}, \quad y_i^1 \in \{0, 1\}, p_i \in [0, 1] \quad (2.1)$$

that is, the Likelihood function is *Bernoulli*. A reasonable prior for each p_i is to assume that it follows *Beta(1,1)* distribution. So, p_i takes values uniformly in interval $[0, 1]$, which means that the *a priori estimate* of p is given by its mean value 0.5. Then, using *Bayes' theorem*, we can update our knowledge about the unknown value p_i given the observed value y_i^1 after the first trial by multiplying the *Bernoulli Likelihood* (2.1) with a *Beta prior* defined by (2.2):

$$\pi(p_i|\alpha, \beta) = \frac{p_i^{\alpha-1} (1 - p_i)^{\beta-1}}{B(\alpha, \beta)}, \quad (2.2)$$

where $B(\alpha, \beta)$ denotes a Beta function with parameters $\alpha = \beta = 1$. This leads to the *posterior probability*

$$\pi(p_i|y_i^1) \propto L(y_i^1|p_i) \cdot \pi(p_i) = p_i^{y_i^1} (1 - p_i)^{1-y_i^1}, \quad (2.3)$$

which is a *conjugate Beta posterior* with parameters $1 + y_i, 2 - y_i$; i.e. $p_i \sim \text{Beta}(1 + y_i, 2 - y_i)$. After the second trial, we update the estimate of the

latent probability value p_i given the observed data y_i^1, y_i^2 up to trial 2:

$$\pi(p_i|y_i^1, y_i^2) \propto L(y_i^1, y_i^2|p_i) \cdot \pi(p_i) = L(y_i^2|p_i) \cdot \pi(p_i|y_i^1), \quad (2.4)$$

that is, the posterior at trial 1 becomes prior at trial 2. Now, we can apply the same principal to estimate the latent probability value p_i after j independent trials. Denoting (y^1, \dots, y^j) the observed binary lattices up to trial j , we apply *sequential Bayesian analysis* to sequentially update the lattice value p_i after trial j :

$$\begin{aligned} \pi(p_i|y_i^1) &\propto L(y_i^1|p_i) \cdot \pi(p_i), \\ \pi(p_i|y_i^1, y_i^2) &\propto L(y_i^1|p_i) \cdot L(y_i^2|p_i) \cdot \pi(p_i) = L(y_i^2|p_i) \cdot \pi(p_i|y_i^1), \\ &\dots \\ \pi(p_i|y_i^1, \dots, y_i^j) &\propto L(y_i^1|p_i) \cdot \dots \cdot L(y_i^j|p_i) \cdot \pi(p_i) = L(y_i^j|p_i) \cdot \pi(p_i|y_i^1, \dots, y_i^{j-1}). \end{aligned}$$

In other words, a prior distribution at trial j is a posterior distribution at a previous trial $j - 1$. Thus, after 2 updates the posterior distribution for a pixel i can be written as

$$\pi(p_i|y_i^1, y_i^2) \propto \prod_{t=1}^2 L(y_i^t|p_i) \cdot \pi(p_i) = p_i^{y_i^1+y_i^2} \cdot (1-p_i)^{2-(y_i^1+y_i^2)}, \quad (2.5)$$

which is a Beta distribution with parameters $1+y_i^1+y_i^2, 3-(y_i^1+y_i^2)$. Writing the posterior after j independent trials in the same way:

$$\pi(p_i|y_i^1, \dots, y_i^j) \propto \prod_{t=1}^j L(y_i^t|p_i) \cdot \pi(p_i) = p_i^{\sum_{t=1}^j y_i^t} \cdot (1-p_i)^{j-\sum_{t=1}^j y_i^t}, \quad (2.6)$$

we find that the posterior probability has a $Beta(1 + s, 1 + j - s)$ distribution where $s = \sum_{t=1}^j y_i^t$ is the number of correct responses after trial j . Hence, the estimate of a latent probability value p_i after j independent trials is given by $(1 + s)/(2 + j)$. Having estimated the value p_i for every pixel i , we obtain $\pi(p|y^1, \dots, y^j)$ - the *posterior probability map* of lattice p given the observed data lattices (y^1, \dots, y^j) up to trial j . Now we can threshold posterior probability map values at level e.g. 0.95 and 0.05, that is, we classify as 'important' all pixels with posterior probability map values of 0.95 or higher and as 'unimportant' all pixels with posterior probability map values of 0.05 or lower.

2.1 *Original Bubbles with Adaption*

Currently, to achieve a correct classification with the original Bubbles (Gosselin & Schyns 2001), a large number of trials is required (Gosselin & Schyns 2001). The cost of each trial is prohibitive high. An adaptive technique introduced below, in the following referred to as the *original Bubbles with Bayesian adaption*, aims to address this problem by placing the sampling approach in a more logical and statistical setting.

The original Bubbles samples pixels randomly, allowing thereby each pixel to be included in a Bubbles sample. A simple alternative to the exhaustive

sampling procedure would be the adaptive sampling approach which suggests to classify pixels as 'important' or 'unimportant' by thresholding their posterior probability values at level 0.95 and 0.05, respectively. We remove these pixels from further sampling and apply a weighted sampling scheme to the remaining pixels:

$$\begin{aligned} w_i &\propto p_i \\ \sum_{i=1}^n w_i &= 1 \end{aligned} \tag{2.7}$$

i.e. pixel i will be sampled in a further trial with the weight w_i proportional to its posterior probability value and the sum of all weights is equal to 1. A higher weight means that the pixel should have larger chance to be sampled.

2.2 *Drawbacks and Further Improvement*

The problem of ineffective sampling in the exhaustive Bubbles approach can be solved by means of introducing a rule for excluding already classified pixels from further sampling, as described in section (2.1). However, the Bayesian Bubbles with adaption assumes that there is there no spatial dependence in the image. This assumption is implausible, since image pixels are clearly dependent. In the following, we aim to introduce an alternative model which allow to incorporate spatial dependence in the image. Chapters 3 and 4 in-

introduce the *adaptive Bubbles with Beta MRF prior* and the *adaptive Bubbles with Ising prior* which allow spatial dependency to be modeled a priori via Markov random fields (MRFs).

3. ADAPTIVE BUBBLES WITH BETA MRF PRIOR

This chapter introduces the *adaptive Bubbles with Beta MRF prior* which allows to model spatial dependence in the image using Markov random field (MRF) theory. MRFs allow identification of significantly informative image regions which are used for further sampling and to exclude from sampling those image regions which contribute the least to solving our categorization problem. In the following, the Bubbles approach which allows spatial dependency to be modeled via *Beta MRF* with incorporated adaptive algorithm for excluding already revealed pixels from further sampling, will be referred to as the *adaptive Bubbles with Beta MRF prior*. The essential part of the procedure consists of designing an *Markov Chain Monte Carlo (MCMC)* algorithm needed for identification of 'important' sampling regions.

This chapter is organized as follows: section 3.1 provides theory on MRFs and MCMC needed for implementation the algorithm. The adaptive Bubbles with Beta MRF prior procedure is described in section 3.2. To start the procedure running, we have to model the joint distribution and the likelihood, as

discussed in sections 3.3-3.6. Finally, simulated data examples are presented in section 3.7.

3.1 Markov Random Fields and MCMC

Markov Random Fields (MRFs), first introduced in (Besag 1974) and (Geman & Geman 1984), play an important role in spatial statistics. In particular, they are widely used as prior distributions in Bayesian analysis. MRFs can be regarded as an extension of *Markov chains (MC)* where each random variable X_i of the sequence $\{X_1, X_2, \dots\}$ possesses the *Markov property*, namely that for any given present state the future state of the chain does not depend on the past, that is,

$$\mathbb{P}(X_{n+1} = x | X_n = x_n, \dots, X_1 = x_1) = \mathbb{P}(X_{n+1} = x | X_n = x_n). \quad (3.1)$$

MCMC algorithm can be regarded as a Monte Carlo integration using Markov chains. *Monte Carlo* is a conditional simulation algorithm that generates random samples from a given probability distribution. *Markov Chain Monte Carlo* draws these samples by running Markov chains. The way to construct Markov chains includes the *Gibbs sampler* (Geman & Geman 1984) and the *Metropolis-Hasting* approach (Metropolis, Rosenbluth, Rosenbluth, Teller & Teller 1953).

3.1.1 The Metropolis-Hasting Algorithm

The *Metropolis-Hasting algorithm* was introduced by (Hastings 1970) and generalized by (Metropolis et al. 1953). Suppose, given a partition of a state vector into components $x = (x_1, \dots, x_n)$, we aim to update the i^{th} component. Therefore, we have to draw samples from a distribution $\pi(x)$, which we refer to as the *target distribution*. The Metropolis-Hasting algorithm generates a sequence of draws from this distribution in the following way:

1. Consider the i^{th} component x_i of a configuration $x = (x_1, \dots, x_i, \dots, x_n)$.
2. Propose a new value y_i from some distribution $q(x, y)$. Thereby, configuration $y = (x_1, \dots, y_i, \dots, x_n)$ matches configuration x on all but i^{th} position. Distribution $q(x, y)$ is also referred to as a *proposal distribution*.
3. We accept the new configuration $y = (x_1, \dots, y_i, \dots, x_n)$ with a probability

$$\alpha(x, y) = \min \left\{ 1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)} \right\}, \quad (3.2)$$

that is, we generate a uniformly $(0, 1)$ -distributed random variable u and accept a configuration y if $u < \alpha(x, y)$. Otherwise, we keep a configuration x . In the following, this step of algorithm is referred to as the *Metropolis update*.

The Metropolis algorithm considers only symmetric proposals of the form $q(x, y) = q(y, x)$ for all x and y , see (Metropolis et al. 1953) and thus, the acceptance probability $\alpha(x, y)$ defined by (3.2) reduces to

$$\alpha(y|x) = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \right\}. \quad (3.3)$$

3.1.2 The Gibbs Sampler

The *Gibbs update* can be regarded as a special case of a Metropolis update. Suppose, as above, that we want to update the i^{th} element of a configuration x . We consider a proposal distribution defined in the following way:

$$q(x, y) = \begin{cases} \pi(y_i|x_{\setminus i}) & y_{\setminus i} = x_{\setminus i} \text{ for } i = 1, \dots, n \\ 0 & \text{otherwise} \end{cases}$$

Thereby, $\pi(y|x_{\setminus i})$ denotes a conditional distribution of the proposed configuration y given all but one component of a given configuration x , that is,

$$x_{\setminus i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n).$$

With this proposal distribution, the corresponding acceptance probability is obtained by

$$\begin{aligned} \alpha(x, y) &= \min \left\{ 1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)} \right\} \\ &= \min \left\{ 1, \frac{\pi(y)/\pi(y_i|x_{\setminus i})}{\pi(x)/\pi(x_i|y_{\setminus i})} \right\} \\ &= \min \left\{ 1, \frac{\pi(y)/\pi(y_i|y_{\setminus i})}{\pi(x)/\pi(x_i|x_{\setminus i})} \right\} \quad \text{since } y_{\setminus i} = x_{\setminus i}. \end{aligned} \quad (3.4)$$

By definition of the conditional probability we have:

$$\begin{aligned}\pi(y_i|y_{\setminus i}) &= \frac{\pi(y_i, y_{\setminus i})}{\pi(y_{\setminus i})} = \frac{\pi(y)}{\pi(y_{\setminus i})} \\ \pi(x_i|x_{\setminus i}) &= \frac{\pi(x_i, x_{\setminus i})}{\pi(x_{\setminus i})} = \frac{\pi(x)}{\pi(x_{\setminus i})}\end{aligned}\quad (3.5)$$

and thus, substituting (3.5) in (3.4), the acceptance probability reduces to

$$\alpha(x, y) = \min \left\{ 1, \frac{\pi(y_{\setminus i})}{\pi(x_{\setminus i})} \right\} = 1 \quad \text{since } y_{\setminus i} = x_{\setminus i}. \quad (3.6)$$

Thus, (3.6) indicates that for the proposed configuration y which matches the given configuration x on all but i^{th} component, all proposed distributions are automatically accepted.

3.2 Adaptive Bubbles with Beta MRF Prior: the Procedure

In this section we consider the adaptive Bubbles with Beta MRF prior, assuming that the latent image p consists of pixels with values in interval $[0, 1]$ corresponding to the probability that the pixel is important. Thus, we aim to make inference about unknown probability values p . For modeling probabilities, the *Beta distribution* is an obvious choice to use and here, to incorporate spacial dependence in the image, we use a *Beta Markov Random Field* as a prior distribution. Beta MRFs are simple to construct: in analogy to Gaussian MRFs where a (finite-dimensional) random vector follows a multivariate normal (or Gaussian) distribution, Beta MRFs assume a random vector to

be Beta distributed.

Conditioned on the *observed binary lattices* (y^1, \dots, y^j) up to trial j , we can evaluate the *posterior distribution* for hidden probabilities p in a typical Bayesian way:

$$\pi(p|y^1, \dots, y^j) \propto L(y^1, \dots, y^j|p) \cdot \pi(p), \quad (3.7)$$

where $L(y^1, \dots, y^j|p)$ is a *likelihood function* and a *prior distribution* $\pi(p)$ is given by a *Beta MRF*. Thus, in order to evaluate the posterior $\pi(p|y^1, \dots, y^j)$ from (3.7), we require knowledge of the joint distribution $\pi(p)$ and the likelihood $L(y^1, \dots, y^j|p)$. The following two sections deal with its modeling. Therefore, we first derive the result for one image pixel i and then expand it to the whole image.

3.3 Modeling the joint distribution.

3.3.1 For one image pixel

In order to evaluate the joint distribution $\pi(p)$ we will use the fact that the full-conditional distribution $\pi(p_i|p_{\setminus i})$ is proportional to the joint distribution, i.e. up to a normalizing constant it holds:

$$\pi(p_i|p_{\setminus i}) = \frac{\pi(p_i, p_{\setminus i})}{\pi(p_{\setminus i})} \propto \pi(p). \quad (3.8)$$

Thus, we have to make an assumption on the form of the distribution $\pi(p_i|p_{\setminus i})$.

In the Gaussian case (Rue & Held 2005), $\pi(p_i|p_{\setminus i})$ is assumed to be normal:

$$\pi(p_i|p_{\setminus i}) \propto \exp \left\{ -\frac{1}{2}(p_i)^2 Q_{ii} - p_i \sum_{k \sim i} Q_{ik} p_k \right\}, \quad (3.9)$$

matrix $Q = \Sigma^{-1}$ given by the inverse of a covariance matrix Σ , is a so-called *precision matrix*. In (3.9) $i \sim k$ means that pixel i is a neighbor of pixel k . In the case of Beta MRF we assume that $\pi(p_i|p_{\setminus i})$ follows a Beta distribution:

$$\pi(p_i|p_{\setminus i}) \propto (p_i)^{\{\alpha-1\}} \cdot (1-p_i)^{\{\beta-1\}} \cdot \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \quad (3.10)$$

where α and β are model parameters controlling the shape of the density function of Beta distribution and $\Gamma(\cdot)$ is the gamma function. Recall, that the density of the Beta distribution is given by

$$\begin{aligned} f(p, \alpha, \beta) &= \frac{p^{\alpha-1}(1-p)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1} du} \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} = \frac{1}{B(\alpha, \beta)} p^{\alpha-1}(1-p)^{\beta-1} \end{aligned} \quad (3.11)$$

with $\Gamma(\cdot)$ and $B(\cdot)$ denoting the *Gamma* and the *Beta* function, respectively.

However, the above expression (3.10) used for modeling $\pi(p_i|p_{\setminus i})$ does not take into account any spatial dependence among image pixels. To incorporate spatial dependence in the image, we extend model (3.10) in a following way:

$$\begin{aligned} \pi(p_i|p_{\setminus i}) &\propto (p_i)^{\{\alpha - \sum_{k \sim i} \theta \cdot \log(1-p_k)\}} \cdot (1-p_i)^{\{\beta - \sum_{k \sim i} \theta \cdot \log(p_k)\}} \\ &\times \frac{\Gamma\{\alpha - \sum_{k \sim i} \theta \cdot \log(1-p_k) + 1\} \Gamma\{\beta - \sum_{k \sim i} \theta \cdot \log(p_k) + 1\}}{\Gamma\{\alpha + \beta - \sum_{k \sim i} \theta \cdot \log(1-p_k) - \sum_{k \sim i} \theta \cdot \log(p_k) + 2\}}, \end{aligned} \quad (3.12)$$

by introducing one additional model parameter θ . It controls the strength of the dependence among neighboring pixels i and k by entering the summations $\sum_{k \sim i} \theta \cdot \log(1 - p_k)$ and $\sum_{k \sim i} \theta \cdot \log(p_k)$. If $\theta = 0$, these terms become equal zero and thus, there is no dependence among image pixels. High values of θ lead to non-zero summation terms and thus, neighboring pixels become clearly dependent. Possible realizations of Beta MRFs obtained by running the Metropolis algorithm with Gibbs update and different values of the parameters α , β and θ are represented in Figure 3.1. Using the notation

$$\begin{aligned} A_{i,1} &= \alpha - \sum_{k \sim i} \theta \cdot \log(1 - p_k) \text{ and} \\ A_{i,2} &= \beta - \sum_{k \sim i} \theta \cdot \log(p_k), \end{aligned}$$

(3.12) becomes:

$$\pi(p_i | p_{\setminus i}) \propto (p_i)^{\{A_{i,1}\}} \cdot (1 - p_i)^{\{A_{i,2}\}} \cdot \frac{\Gamma(A_{i,1} + 1)\Gamma(A_{i,2} + 1)}{\Gamma(A_{i,1} + A_{i,2} + 2)}. \quad (3.13)$$

Taking logarithm on both sides of the equation (3.13), leads to the following logarithmic model:

$$\begin{aligned} \log \{ \pi(p_i | p_{\setminus i}) \} &\propto \left\{ \alpha - \sum_{k \sim i} \theta \cdot \log(1 - p_k) \right\} \cdot \log(p_i) \\ &+ \left\{ \beta - \sum_{k \sim i} \theta \cdot \log(p_k) \right\} \cdot \log(1 - p_i) \\ &+ \log \{ \Gamma(A_{i,1} + 1) \} + \log \{ \Gamma(A_{i,2} + 1) \} - \log \{ \Gamma(A_{i,1} + A_{i,2} + 2) \}. \end{aligned} \quad (3.14)$$

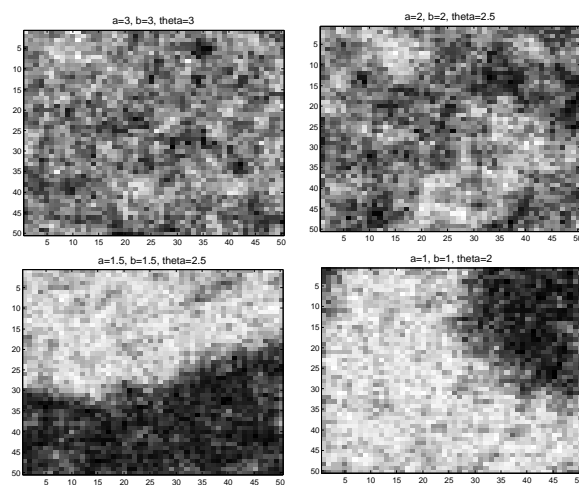


Fig. 3.1: Some realizations of Beta Markov Random Field after 1000 MCMC updates obtained by running the Metropolis algorithm with Gibbs update and parameters $\alpha = 3.0$, $\beta = 3.0$, $\theta = 3.0$ (upper left), $\alpha = 2.0$, $\beta = 2.0$, $\theta = 2.5$ (upper right), $\alpha = 1.5$, $\beta = 1.5$, $\theta = 2.5$ (lower left) and $\alpha = 1.0$, $\beta = 1.0$, $\theta = 2.0$ (lower right).

3.3.2 For the whole image

Since modeling of a joint distribution $\pi(p)$ can only be possible up to a normalizing constant, for its evaluation, it makes sense to define a function which is equal to a joint probability only up to a constant as well. Choosing a particular value $p^* = \{p_1^*, \dots, p_n^*\}$ and assuming that $\pi(p^*)$ is finite, we can define a *negpotential function* after trial j as follows:

$$Q(p) = \log \left\{ \frac{\pi(p)}{\pi(p^*)} \right\}. \quad (3.15)$$

In (Besag 1974), without loss of generality p^* has been chosen to be $\mathbf{0} = \{0, \dots, 0\}$. Several results for MRFs were proved by (Besag 1974) using $Q(p)$ with $p^* = 0$. These results will hold for any arbitrary choice of a value p^* (Kaiser & Cressie 2000). From (3.15) we conclude that the knowledge of $Q(p)$ is equivalent to the knowledge of $\pi(p)$, that is, up to a normalizing constant, the joint distribution can be written as:

$$\pi(p) \propto \exp\{Q(p)\} \quad (3.16)$$

or, taking a logarithm:

$$\log\{\pi(p)\} \propto Q(p). \quad (3.17)$$

In the following, we aim to write $Q(p)$ using the *log-linear models* (Darroch, Lauritzen & Speed 1980), expanding the $\log\{\pi(p)\}$ as:

$$\log\{\pi(p)\} = u_\phi + \sum_i u_i(p_i) + \dots + u_{1,\dots,n}(p_1, \dots, p_n). \quad (3.18)$$

Then, it can be shown (Besag 1974) that the negpotential function can be written in the following form:

$$\begin{aligned} Q(p) &= \sum_{i=1}^n H_i(p_i) + \sum_{1 \leq i < k \leq n} H_{i,k}(p_i, p_k) \\ &+ \sum_{1 \leq i < k < m \leq n} H_{i,k,m}(p_i, p_k, p_m) + \dots + \\ &+ H_{1,2,\dots,n}(p_1, p_2, \dots, p_n). \end{aligned} \quad (3.19)$$

Thus, to evaluate the function $Q(p)$, we will have to define the H -functions. (Kaiser & Cressie 2000) provide an alternative formulation of a well known Hammersley-Clifford theorem, showing that any function $H_{i,k,\dots,r}$ is equal to 0 unless the pixels $\{i, k, \dots, r\}$ form a clique. Thereby, the clique is defined by pixels or a set of pixels such that each pixel is contained in the set of neighbors of every other pixel in a set. For the sake of simplicity, we restrict our expansion to the case when only first and second order H -functions enter the summation (3.19). In other words, we use the first order $H_i(p_i)$ -function corresponding to a single point i and the second order $H_{i,k}(p_i, p_k)$ -function corresponding to its nearest neighbors given by the four connections between points i and k . We ignore those H -functions having the order 3 or higher. In

(Kaiser & Cressie 2000) is shown that H-functions of the first and the second order can be evaluated as

$$H_i(p_i) = \log \left[\frac{\pi_i(p_i | \{p_k^*, k \neq i\})}{\pi_i(p_i^* | \{p_k^*, k \neq i\})} \right] \quad (3.20)$$

respectively

$$H_{i,k}(p_i, p_k) = \log \left[\frac{\pi_i(p_i | p_k, \{p_m^*, m \neq i, k\})}{\pi_i(p_i^* | p_k, \{p_m^*, m \neq i, k\})} \cdot \frac{\pi_i(p_i^* | \{p_m^*, m \neq i\})}{\pi_i(p_i | \{p_m^*, m \neq i\})} \right]. \quad (3.21)$$

The full conditional p.d.f. of the Beta distribution (conditioned on the set of neighboring points) can be expressed in terms of an exponential family structure:

$$\pi_i(p_i | p_{\setminus i}) = \exp \left\{ \sum_{m=1}^2 A_{i,m}(p_{\setminus i}) T_m(p_i) - B_i(p_{\setminus i}) + C_i(p_i) \right\} \quad (3.22)$$

where parameter functions $A_{i,m}(p_{\setminus i})$ are given by:

$$\begin{aligned} A_{i,1}(p_{\setminus i}) &= \alpha - \sum_{k \sim i} \theta \log(1 - p_k) \\ A_{i,2}(p_{\setminus i}) &= \beta - \sum_{k \sim i} \theta \log(p_k) \end{aligned} \quad (3.23)$$

with sufficient statistics defined by:

$$T_1(p_i) = \log(p_i) \text{ respectively } T_2(p_i) = \log(1 - p_i).$$

Further, parameter functions $B_i(p_{\setminus i})$ and $C_i(p_i)$ can be written as

$$\begin{aligned} B_i(p_{\setminus i}) &= \log\{\Gamma(A_{i,1}(p_{\setminus i}) + 1)\} + \log\{\Gamma(A_{i,2}(p_{\setminus i}) + 1)\} \\ &\quad - \log\{\Gamma(A_{i,1}(p_{\setminus i})A_{i,2}(p_{\setminus i}) + 2)\} \end{aligned} \quad (3.24)$$

and $C_i(p_i) = 0$.

Substitution of (3.22) in (3.20) and (3.21) leads to:

$$H_i(p_i) = \sum_{m=1}^2 [A_{i,m}(p_{\setminus i}^*) \{T_m(p_i) - T_m(p_i^*)\}] + C_i(p_i) - C_i(p_i^*) \quad (3.25)$$

respectively

$$H_{i,k}(p_i, p_k) = \sum_{m=1}^2 \left[\{A_{i,m}(p_k, p_{\setminus i \setminus k}^*) - A_{i,m}(p_{\setminus i}^*)\} \cdot \{T_m(p_i) - T_m(p_i^*)\} \right]. \quad (3.26)$$

H -function of the first and the second order can be obtained by substituting of (3.23) into the above expressions:

$$\begin{aligned} H_i(p_i) &= \{\alpha - \theta \log(1 - p_k^*)\} \{\log(p_i) - \log(p_i^*)\} \\ &+ \{\beta - \theta \log(p_k^*)\} \{\log(1 - p_i) - \log(1 - p_i^*)\} \end{aligned} \quad (3.27)$$

respectively

$$\begin{aligned} H_{i,k}(p_i, p_k) &= - \theta [\{\log(1 - p_k) - \log(1 - p_k^*)\} \cdot \{\log(p_i) - \log(p_i^*)\}] \\ &- \theta [\{\log(p_k) - \log(p_k^*)\} \cdot \{\log(1 - p_i) - \log(1 - p_i^*)\}]. \end{aligned} \quad (3.28)$$

Then, by substituting (3.27) and (3.28) into the $Q(p)$ -expansion given by (3.19), we come up with a following expression for the negpotential function:

$$\begin{aligned} Q(p) &= \sum_{i=1}^n \{\alpha \log(p_i) + \beta \log(1 - p_i)\} \\ &\quad - \sum_{1 \leq i < k \leq n} \theta \{\log(p_i) \log(1 - p_k) + \log(1 - p_i) \log(p_k)\}. \end{aligned} \quad (3.29)$$

Thus, using (3.17), the logarithm of the prior can be written as:

$$\begin{aligned} \log\{\pi(p)\} &\propto \sum_{i=1}^n \{\alpha \log(p_i) + \beta \log(1 - p_i)\} \\ &\quad - \sum_{1 \leq i < k \leq n} \theta \{\log(p_i) \log(1 - p_k) + \log(1 - p_i) \log(p_k)\} \\ &= \sum_{i=1}^n \left[\left\{ \alpha - \sum_{k \sim i} \theta \log(1 - p_k) \right\} \log(p_i) + \left\{ \beta - \sum_{k \sim i} \theta \log(p_k) \right\} \log(1 - p_i) \right] \end{aligned} \quad (3.30)$$

or, using the notation (3.23), we can write:

$$\log\{\pi(p)\} \propto \sum_{i=1}^n \{A_{i,1} \cdot \log(p_i) + A_{i,2} \cdot \log(1 - p_i)\}. \quad (3.31)$$

3.4 Modeling the Likelihood.

To evaluate the likelihood from (3.7), we assume that conditioned on p all y_i are independent. Thus, we can write the likelihood after trial j in the following way:

$$L(y^1, \dots, y^j | p) = \prod_{i=1}^n (p_i)^{\sum_{t=1}^j y_i^t} \cdot (1 - p_i)^{N_i^j - \sum_{t=1}^j y_i^t}, \quad (3.32)$$

where $\sum_{t=1}^j y_i^t$ is the number of times pixel i has been classified as 'important' after j trials and N_i^j is the total number of times pixel i has been visited (revealed to observer) after j trials. Taking the logarithm of (3.32) on both sides, we obtain the log-likelihood for the whole image after trial j :

$$l(y^1, \dots, y^j | p) = \sum_{i=1}^n \left\{ \sum_{t=1}^j y_i^t \cdot \log(p_i) + (N_i^j - \sum_{t=1}^j y_i^t) \cdot \log(1 - p_i) \right\} \quad (3.33)$$

3.5 Modeling posterior distribution.

We can evaluate the logarithm of posterior $\log\{\pi(p|y^1, \dots, y^j)\}$ by taking the logarithm of (3.7), that is, by substituting the expression (3.33) for the log-likelihood and the expression (3.30) for the log-prior in the following equation:

$$\log\{\pi(p|y^1, \dots, y^j)\} \propto l(y^1, \dots, y^j | p) + \log\{\pi(p)\}. \quad (3.34)$$

Thus, we can evaluate a posterior distribution $\pi(p|y^1, \dots, y^j)$ as follows:

$$\begin{aligned} \log\{\pi(p|y^1, \dots, y^j)\} &\propto \sum_{i=1}^n \left[\sum_{t=1}^j y_i^t + \alpha - \sum_{k \sim i} \theta \log(1 - p_k) \right] \cdot \log(p_i) \\ &+ \sum_{i=1}^n \left[(N_i^j - \sum_{t=1}^j y_i^t) + \beta - \sum_{k \sim i} \theta \log(p_k) \right] \cdot \log(1 - p_i). \end{aligned} \quad (3.35)$$

or, using the notation,

$$\begin{aligned} A_{i,1} &= \alpha - \sum_{k \sim i} \theta \log(1 - p_k) \\ A_{i,2} &= \beta - \sum_{k \sim i} \theta \log(p_k), \end{aligned}$$

we obtain:

$$\log\{\pi(p|y^1, \dots, y^j)\} \propto \sum_{i=1}^n \left[\left\{ \sum_{t=1}^j y_i^t + A_{i,1} \right\} \cdot \log(p_i) + \left\{ (N_i^j - \sum_{t=1}^j y_i^t) + A_{i,2} \right\} \cdot \log(1 - p_i) \right]. \quad (3.36)$$

3.6 The Metropolis-Hasting Algorithm

For a possible update of lattice p after trial j , we will iterate the MCMC algorithm many times, applying the following Metropolis-Hasting update to each pixel i :

$$\begin{aligned} \alpha(\tilde{p}_i|p_i) &= \min \left\{ 1, \frac{\pi(\tilde{p}_i|y_i^1, \dots, y_i^j)}{\pi(p_i|y_i^1, \dots, y_i^j)} \right\} \\ &= \min \left\{ 1, \frac{L(y_i^1, \dots, y_i^j|\tilde{p}_i)}{L(y_i^1, \dots, y_i^j|p_i)} \cdot \frac{\pi(\tilde{p}_i|p_{\setminus i})}{\pi(p_i|p_{\setminus i})} \right\}. \end{aligned} \quad (3.37)$$

$\alpha(\tilde{p}_i|p_i)$ defines the probability with which the new proposed lattice \tilde{p} will be accepted. The logarithmic ratio is then given by:

$$\begin{aligned}
& \log\left\{\frac{L(y_i^1, \dots, y_i^j|\tilde{p}_i)}{L(y_i^1, \dots, y_i^j|p_i)} \cdot \frac{\pi(\tilde{p}_i|p_{\setminus i})}{\pi(p_i|p_{\setminus i})}\right\} \\
& + \sum_{t=1}^j y_i^t \cdot \{\log(\tilde{p}_i) - \log(p_i)\} + (N_i^j - \sum_{t=1}^j y_i^t) \cdot \{\log(1 - \tilde{p}_i) - \log(1 - p_i)\} \\
& + \left\{\alpha - \sum_{i \sim k} \theta \cdot \log(1 - p_k)\right\} \{\log(\tilde{p}_i) - \log(p_i)\} \\
& + \left\{\beta - \sum_{i \sim k} \theta \cdot \log(p_k)\right\} \{\log(1 - \tilde{p}_i) - \log(1 - p_i)\}. \tag{3.38}
\end{aligned}$$

Thus, we run through all the lattice points in turn updating the lattice p by \tilde{p} . If the logarithm $\log(u)$ of the uniformly $[0, 1]$ distributed random variable u does not exceed $\log\{\alpha(\tilde{p}_i|p_i)\}$, we accept the proposed lattice value \tilde{p} , otherwise we keep the previous lattice p .

3.7 Simulated Experiments.

In this section we deal with a situation when the human observer is not present in the experiment and thus, we have to replicate the Bubbles situation. The aim is to correctly find the true hidden values of lattice p by generating a sequence of trials. The trial consists of sampling a certain proportion of image pixels in order to replicate the Bubbles image. We will simulate the behavior of the observer by assigning labels 'important' or 'unimportant' to image pixels. This will be done in a stochastic way for each pixel revealed in

a trial.

3.7.1 Replicating Bubbles situation: Data generation and simulation behavior of the observer

At this stage of the procedure we have to come up with some mechanism for generating the observed data $y_i^j \in \{0, 1\}$ for each pixel i (in a trial j) and compute values N_i^j and $\sum_{t=1}^j y_i^t$ required for calculating the logarithmic ratio (3.38). This can be done by iterating the algorithm summarized below:

1. In each trial we reveal a certain portion of image pixels (e.g. 5% or 10%) and calculate the proportion of object pixels among all revealed pixels.
2. For each pixel i we calculate the probability of correct classification pcc_i , i.e. the probability to classify pixel i correctly as 'important' or 'unimportant'. At this stage of the procedure we simulate the behavior of the observer using the proposed classification pattern which we set in advance. We define pcc_i for each pixel i using a c.d.f. pattern of the $Beta(\alpha, \beta)$ distribution, see Figure 3.2: the x -axis represents proportions of object pixels among all revealed pixels in a trial and the values of the c.d.f. on the y -axis give us the required probabilities of correct classification pcc_i .

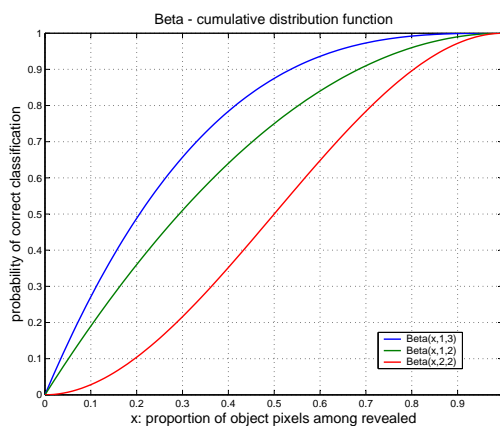


Fig. 3.2: Cumulative distribution function (c.d.f.) of Beta - distribution is used as a classification pattern: the x -axis represents the proportion of object pixels among all revealed and the values of the c.d.f. correspond to the probability of correct classification.

3. Now, with probability pcc_i we classify every pixel i correctly, i.e. pixel i adopts the 'right' color: object pixel is converted to 'black' and non-object pixel to 'white'. We misclassify pixel i with probability $(1-pcc_i)$, i.e. pixel i adopts the 'wrong' color: object pixel is converted to 'white' and non-object pixel to 'black'.

Though value N_i increases by 1 every time we visit pixel i , value $\sum_{t=1}^j y_i^t$ increases by 1 only if after visiting pixel i , we classify it as an object pixel.

The MCMC Step

The adaptive Bubbles with Beta MRF prior is an iterative procedure which consists in iterating two following algorithm steps. The first step of the procedure consists in generating data and simulating behavior of the observer, as described above. The second step consists of running the MCMC algorithm to identify 'important' regions from which to sample from further in step 1 of the procedure. The number of MCMC iteration as well as tuning parameters of the MCMC procedure will be specified in section 3.7.2 below.

3.7.2 Tuning Parameters

To start the procedure running, we have to specify tuning parameters of the experiment which includes the following:

1. Defining the number of image pixels revealed in each trial and setting parameters of Beta-distribution used to simulate the behavior of the observer.
2. Setting parameters α and β and θ of the Beta MRF.
3. Specifying 'very high' and 'very low' values of posterior probability map used as threshold values for converting image pixels to 'black' or 'white'.

Note, that the first point specifies parameters of the data generation procedure whereas the second and the third points refer to the actual tuning parameters of the MCMC procedure.

Choosing parameters of Beta-distribution and setting the number of image pixels revealed in each trial, the size of the object has to be taken into consideration. If the number of pixels revealed in each trial is too small compared to the total size of the image, the proportion of object pixels among all revealed pixels will be small as well. Small values of the proportion will lead to a small probability of correct classification and thus, to higher misclassification rates. In our simulated experiments, we reveal 10% of image pixels every trial. Further, we use $Beta(1,3)$ - c.d.f. to simulate the behavior of the observer in the first example, where the proportion of object pixels in the image accounts to 24%, see Figure 3.3. In the second example, see Figure 3.6, the proportion of object pixels in the image is approximately 38% and, thus, higher values for the proportion of object pixels among all revealed pixels can be obtained. In this case we use $Beta(1,2)$ - c.d.f. for simulating observer's behavior.

Values 0.95 and 0.05 are used as threshold values for classification of posterior expectation map values as 'very high' or 'very low', respectively. In other words, those pixels which have posterior probability map values of 0.95 or higher will be classified as 'black' and those with values of 0.05 or lower

will be classified as 'white'. In the next step, these classified pixels will be excluded from further sampling.

Finally, the Beta MRF with parameters $\alpha = 2$, $\beta = 2$, $\theta = 2.5$ is used as a prior distribution. Recall that parameters α and β control the smoothness within the image, see Figure 3.1. Choosing parameter values as above allows us to observe the dependence within the image (clusters of black and white pixels), without separating the image into two regions as in case $\alpha = 1$, $\beta = 1$, $\theta = 2$. However, it would be of advantage to adjust the parameter values on-line, i.e. to estimate parameter values after every MCMC run, in order to obtain the best estimates for the posterior probability map, see chapter 6 for details.

Figures 3.4 and 3.7 represent possible realizations of the estimated posterior probability maps for the 'cross' and the 'cameraman' example, respectively. Estimated error rates after different numbers of trials for the *adaptive Bubbles with Beta MRF prior* compared with the *original Bubbles* and the *original Bubbles with Bayesian adaption* are represented in Figures 3.5 for the 'cross' and 3.8 for the 'cameraman'. To calculate the error rate after each update u , we compare the posterior expectation image (obtained by thresholding posterior probability map values at 0.5 level) produced after this update with a

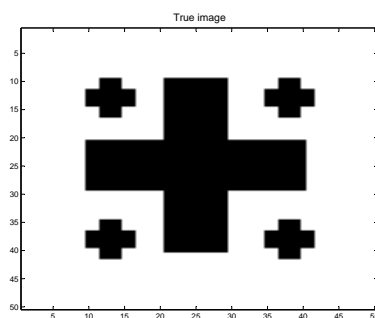


Fig. 3.3: The true 50×50 image. Proportion of object pixels in the image $\approx 24\%$.

We use c.d.f. of the Beta (1, 3) distribution to simulate the behavior of the observer.

true image, that is, we define an error rate $e(u)$ in the following way:

$$e(u) = \frac{\text{number of missclassified pixels after update } u}{\text{total number of pixels in the image}}.$$

We can observe, that the *adaptive Bubbles with Beta MRF prior* as well the *original Bubbles with Bayesian adaption* performs significantly better than the *original Bubbles*. For the *adaptive Bubbles with Beta MRF prior*, after running the MCMC procedure for the first time, we observe rapid decrease in error rates followed by the further moderate decrease, whereas error rates for the *original Bubbles* remain nearly the same across trials. Rapid decrease in error rates for the *original Bubbles with Bayesian adaption* is driven merely by reweighting posterior probability map values and thus, giving pixels with higher weights larger chance to be sampled.

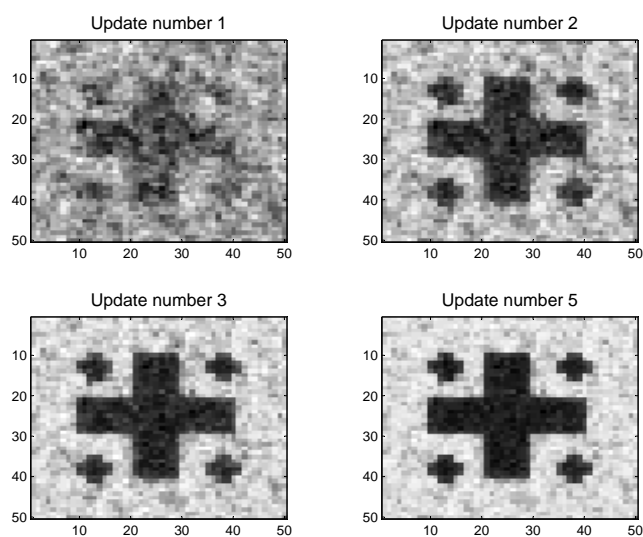


Fig. 3.4: Estimated posterior probability map after different number of updates.

Estimated using Beta MRF with parameters $\alpha = 2$, $\beta = 2$, $\theta = 2.5$.

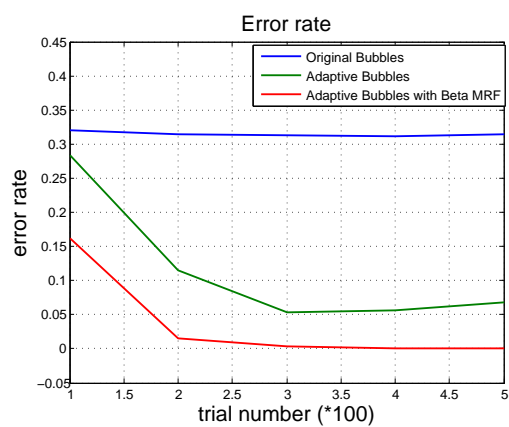


Fig. 3.5: Error rates after different numbers of trials for the cross example: original Bubbles (blue line), original Bubbles with Bayesian adaption (green line) and adaptive Bubbles with Beta MRF prior (red line).

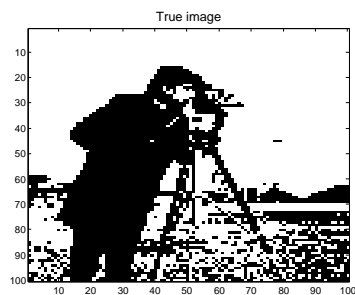


Fig. 3.6: The true 100×100 image. Proportion of object pixels in the image $\approx 38\%$. We use c.d.f. of Beta $(1, 2)$ distribution to simulate behavior of the observer.

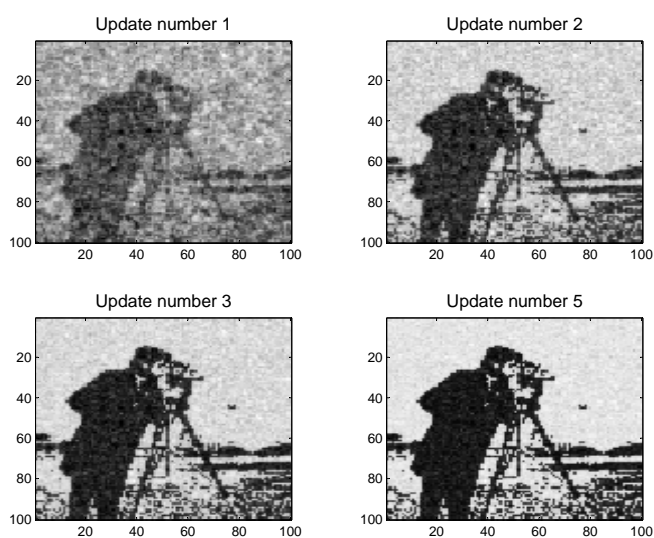


Fig. 3.7: Estimated posterior probability map after different number of updates.

Estimated using Beta MRF with parameters $\alpha = 2$, $\beta = 2$, $\theta = 2.5$.

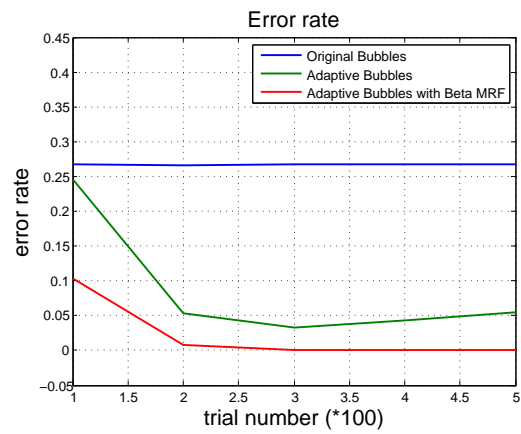


Fig. 3.8: Error rates after different numbers of trials for the cameraman example: original Bubbles (blue line), original Bubbles with Bayesian adaption (green line) and adaptive Bubbles with Beta MRF prior (red line).

4. ADAPTIVE BUBBLES WITH ISING PRIOR

In chapter 3 we assumed that the latent image p consist of values in interval $[0, 1]$ and thus, we used *Beta MRF* in order to make inference about latent probabilities p . Now we assume that the latent image consists of values x having binary $\{-1, +1\}$ structure: $x_i = \{-1\}$ means that the pixel i is 'important' and $x_i = \{+1\}$ means that the pixel i is 'unimportant'. In order to make inference about binary values x , we will use the *Ising model* described below.

This chapter is organized as follows: section 4.1 discusses the Ising model and its generalization, the *autologistic model*. The adaptive Bubbles with Ising prior procedure is described in section 4.2 and applied to the simulated data in section 4.3.

4.1 Ising and Autologistic Models

Here we consider binary spatial data defined on a two-dimensional lattice. We consider a MRF on a rectangular lattice n of dimension $m \times m'$ with the lattice points x_i taking values $\{-1, +1\}$. This is then the *Binary Markov Random Field*, see (Besag 1974). Index $i = 1, \dots, n$ is defined such that the lattice points are ordered from top to bottom in each column and columns from left to right.

The *Ising* model is an example of a MRF, defined as follows:

$$\pi(x|\beta) = \frac{\exp\{\beta U(x)\}}{z(\beta)}, \quad (4.1)$$

where $z(\beta)$ is the *normalizing constant* and the *energy function* $U(x)$ takes the form

$$U(x) = \sum_{i \sim k}^n x_i x_k.$$

Here, $i \sim k$ means that pixel i is a neighbor of pixel k and the clique potential $V_C(x) = x_i x_k$ is simply defined as a product of neighboring pixels values. Using this neighborhood structure, we can write the full conditional distribution for the pixel i in the following form:

$$\pi(x_i|x_{\setminus i}) \propto \exp \left\{ \beta \sum_{i \sim k} x_i x_k \right\}. \quad (4.2)$$

The equivalence between the models (4.2) and (4.1) is given by the Hammersley-Clifford theorem. The parameter β is a smoothing parameter which measures

the dependence within the image, i.e., the strength of the dependence between the neighboring points i and k . $V_C(x)$ is a sufficient statistic for β .

The *autologistic model* proposed by (Besag 1974) is an extension of the Ising model, defined as

$$\pi(x|\beta) = \frac{\exp\{\beta_0 V_0(x) + \beta_1 V_1(x)\}}{z(\beta)}. \quad (4.3)$$

In (4.3), $z(\beta)$ denotes the normalizing constant

$$z(\beta) = \sum_{x_1} \dots \sum_{x_n} \{\beta_0 V_0(x) + \beta_1 V_1(x)\} \quad (4.4)$$

which is extremely difficult to compute in general. Further,

$$V_0(x) = \sum_{i=1}^n x_i \quad \text{and} \quad V_1(x) = \frac{1}{2} \sum_{i \sim k} x_i x_k. \quad (4.5)$$

In (4.5) $V_0(x)$ is the overall sum of the variables. Parameter β_0 represents the abundance of values and β_1 is a smoothing parameter. Positive values for β_0 lead to more $\{+1\}$ ('white') in realizations of x whereas negative values for β_0 lead to realizations of x having more $\{-1\}$ ('black') patches, see figure 4.1. If $\beta_0 = 0$, the autologistic model reduces to the Ising model. The parameter β_1 controls the level of spacial aggregation in the image. If $\beta_1 = 0$, then the pixel values are independent of one another. While positive values of β_1 encourage neighboring pixels to take like values. The constant $1/2$ in the expression for $V_1(x)$ guarantees that each neighboring pair enters the summation only once. Note that most of the lattice points have 4 neighbors:

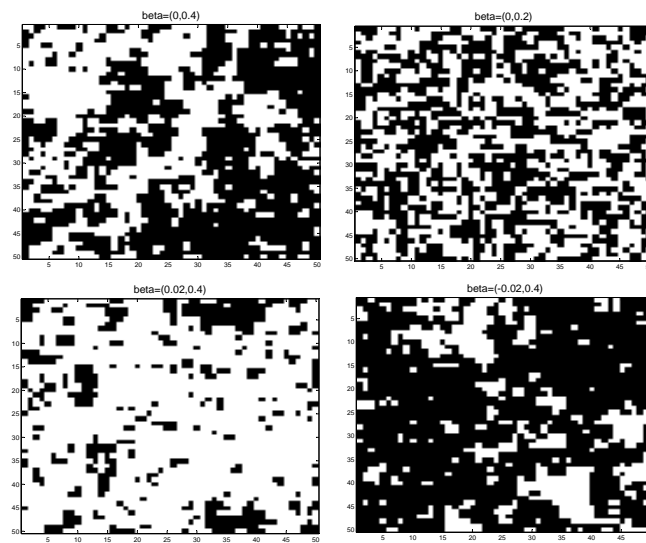
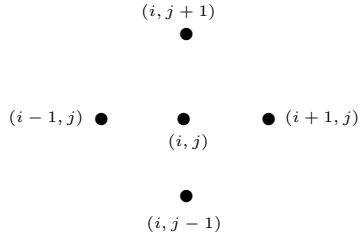


Fig. 4.1: Some realizations of binary Markov Random Field after 1000 MCMC updates with $\beta = (0, 0.4)$ (upper left), $\beta = (0, 0.2)$ (upper right), $\beta = (0.02, 0.4)$ (lower left) and $\beta = (-0.02, 0.4)$ (lower right). Simulated using Ising model.



but along the edges of the lattice each point has either 2 or 3 neighbors. The normalizing constant $z(\beta)$ from (4.3) is generally unknown analytically. Certain computationally and statistically efficient methods for the calculation of the normalizing constant are presented in (Pettitt, Friel & Reeves 2003). Up to a normalizing constant $z(\beta)$ we will use the following notation:

$$\pi(x|\beta) \propto q(x|\beta) = \exp\{\beta_0 V_0(x) + \beta_1 V_1(x)\}. \quad (4.6)$$

It is possible to define a more parameter rich model:

$$\pi(x|\beta) \propto \exp \left\{ \sum_i \beta_i x_i + \sum_{i \sim k} \beta_{ik} x_i x_k \right\} \quad (4.7)$$

where parameters β_i and β_{ik} are not constant and $i \sim k$ means that pixel i is a neighbor of pixel k . But this would result in more parameters than variables. However, for purposes of our studies we concentrate on the parametrization case where β_i and β_{ik} take constant values.

4.2 Adaptive Bubbles with Ising prior: the Procedure

Here we consider a situation when the true hidden data x has a binary $\{-1, +1\}$ structure, corresponding to whether pixel values are 'important' or not. In order to fulfil the primary goal of the Bubbles experiment, that is, to infer latent pixel values x given the observed values of lattice $y \in \{0, 1\}$, we have to evaluate $\pi(x|y^1, \dots, y^j, \beta)$ - the *posterior distribution* of x given all observed lattices (y^1, \dots, y^j) up to trial j . This is done in a typical Bayesian way:

$$\pi(x|y^1, \dots, y^j, \beta) \propto L(y^1, \dots, y^j|x) \cdot \pi(x|\beta), \quad (4.8)$$

where $L(y^1, \dots, y^j|x)$ is the *likelihood function* and $\pi(x|\beta)$ is the *prior distribution*, defined according to the *autologistic model*:

$$\pi(x|\beta) \propto \exp\left\{\beta_0 \sum_{i=1}^n x_i + \frac{1}{2}\beta_1 x_i \sum_{k \sim i} x_k\right\}. \quad (4.9)$$

In order to make inference for a hidden x values, we iterate the MCMC algorithm many times applying the following Metropolis-Hasting update to each pixel i :

$$\alpha(\tilde{x}_i|x) = \min \left\{ 1, \frac{\pi(\tilde{x}_i|y_i^1, \dots, y_i^j, \beta)}{\pi(x_i|y_i^1, \dots, y_i^j, \beta)} \right\} = \min \left\{ 1, \frac{L(y_i^1, \dots, y_i^j|\tilde{x}_i)}{L(y_i^1, \dots, y_i^j|x_i)} \cdot \frac{\pi(\tilde{x}_i|\beta, x_{\setminus i}^j)}{\pi(x_i|\beta, x_{\setminus i}^j)} \right\}, \quad (4.10)$$

where $\alpha(\tilde{x}_i|x)$ defines the probability with which the new proposed lattice \tilde{x} will be accepted after trial j . Thus, we run through all the lattice pixels in

turn updating the lattice x by \tilde{x} . Since x take values in $\{-1, +1\}$, lattice \tilde{x} matches the given x lattice on all but i^{th} component:

$$\tilde{x} = (x_1, \dots, x_{i-1}, -x_i, x_{i+1}, \dots, x_n).$$

The term $x_{\setminus i}$ defines a lattice where the i^{th} component has been omitted:

$$x_{\setminus i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n).$$

In order to compute the acceptance probability defined in (4.10), we require knowledge of the likelihood $L(y^1, \dots, y^j | x)$. Assuming that conditional on x all y_i are independent, we assume that the likelihood can be written in the following way:

$$L(y^1, \dots, y^j | x) = \prod_{i=1}^n (r_i^j)^{\mathbb{1}_{\{x_i=-1\}}} \cdot (1 - r_i^j)^{\mathbb{1}_{\{x_i=+1\}}}, \quad (4.11)$$

that is, the Likelihood has a Beta structure. In (4.11) $r_i^j \in (0, 1)$ denotes the ratio calculated for each pixel i in the following way:

$$r_i^j = \frac{n_i^j}{N_i^j} \quad (4.12)$$

with $n_i^j = \sum_{t=1}^j y_i^t$ denoting the number of times pixel i has been classified as 'important' and N_i^j is the total number of times pixel i has been visited (revealed to the observer) after j trials, see section 4.3 for details. The term $\mathbb{1}_{\{x_i=-1\}}$ in (4.11) denotes an indicator function which becomes equal to 1 if $x_i = -1$ and 0 otherwise.

proposed lattice point	$\frac{L(y_i^1, \dots, y_i^j \tilde{x}_i)}{L(y_i^1, \dots, y_i^j x_i)}$	$\frac{\pi(\tilde{x}_i \beta, x_i)}{\pi(x_i \beta, x_i)}$
$\tilde{x}_i = -1$ ('black')	$\frac{r_i^j}{1-r_i^j}$	$\exp\{-2\beta_0 x_i - \beta_1 x_i \sum_{k \sim i} x_k\}$
$\tilde{x}_i = 1$ ('white')	$\frac{1-r_i^j}{r_i^j}$	$\exp\{2\beta_0 x_i + \beta_1 x_i \sum_{k \sim i} x_k\}$

Tab. 4.1: Likelihood ratio and ratio of priors.

Using the prior distribution and the likelihood defined by (4.9) and (4.11), we can calculate the ratio of priors and the likelihood ratio for each pixel i based on which lattice pixel ($\tilde{x}_i = -1$ if 'black' or $\tilde{x}_i = 1$ if 'white') has been proposed. The results are summarized in Table 4.1. Substituting these ratios in (4.10), we obtain the following probabilities of acceptance for the case if the proposed lattice pixel is 'black' (4.13):

$$\alpha(\tilde{x}|x) = \min \left\{ 1, \frac{r_i^j}{1-r_i^j} \cdot \exp(-2\beta_0 x_i - \beta_1 x_i \sum_{k \sim i} x_k) \right\} \quad \text{if } \tilde{x}_i = -1 \quad (4.13)$$

and for the case if the proposed lattice pixel is 'white' (4.14):

$$\alpha(\tilde{x}|x) = \min \left\{ 1, \frac{1-r_i^j}{r_i^j} \cdot \exp(2\beta_0 x_i + \beta_1 x_i \sum_{k \sim i} x_k) \right\} \quad \text{if } \tilde{x}_i = +1 \quad (4.14)$$

Thus, we accept the lattice \tilde{x} with probability $\alpha(\tilde{x}|x)$, that is, we generate a uniformly $[0, 1]$ -distributed random variable u and accept \tilde{x} if $u < \alpha(\tilde{x}|x)$.

Otherwise, we keep the lattice x .

Iterating the procedure many times allows us to calculate the posterior probability map value for each pixel i , given the observed data y^1, \dots, y^j up to

trial j . This gives the probability that pixel i is 'important' after collecting data y^1, \dots, y^j . It is estimated as the number of times when pixel converts to 'black' to the total number of MCMC iterations after the stationarity has been reached. The posterior expectation for a hidden value x_i after trial j can be obtained by thresholding the corresponding posterior probability map values at 0.5-level. Further, we want to identify those image regions which have 'very high' or 'very low' posterior probability map values (thresholds for these values will be specified in section 4.3) in order to reveal and exclude them from further sampling. Afterwards, we assign to all remaining (not excluded) pixels, a weight: $w_i \propto r_i$ with $\sum_{i=1}^n w_i = 1$. Finally, we return to the first algorithm step, sampling these pixels with weights w_i .

4.3 Simulated Experiments

Here we again deal with a situation when the human observer is not present in experiment and we have to replicate the Bubbles situation. However, in contrast to the experiment where Beta MRF is used as a prior, here we assume that the true hidden data x has a binary structure, and thus, we aim to make inference about binary values $x \in \{-1, +1\}$.

As in the simulated experiment described above, we first generate the observed data $y_i^j \in \{0, 1\}$ for each pixel i and compute corresponding values

r_i^j required for calculating the likelihood (4.11). For each image pixel i we calculate this ratio in the following way:

$$r_i^j = \frac{n_i^j}{N_i^j}, \quad (4.15)$$

where $n_i^j = \sum_{t=1}^j y_i^t$ represents the number of visits to pixel i when i belongs to the object and N_i^j is the total number of visits to pixel i after trial j . The iterative algorithm used to compute the values n_i^j and N_i^j is implemented in the same way as described in section 3.7 for the adaptive Bubbles with Beta MRF prior.

4.3.1 Tuning Parameters

As in the simulated experiments with Beta MRF prior, we reveal 10% of image pixels in every trial and use $Beta(1, 3)$ - c.d.f. to simulate the behavior of the observer in the 'cross' example, where the proportion of object pixels in the image accounts to 24%. In the 'cameraman' example, the proportion of object pixels in the image is approximately 38% and thus, we use $Beta(1, 2)$ - c.d.f. for simulating observer's behavior.

Values 0.95 and 0.05 are used as threshold values for classification of posterior expectation map values as 'very high' or 'very low', respectively.

Finally, we use the autologistic model with parameters $\beta_0 = 0$ and $\beta_1 = 0.4$ (i.e. Ising model) as a prior distribution. Recall that the parameter β_0 con-

trols the relative abundance of object pixels and β_1 is a smoothing parameter which measures the strength of the dependence between neighboring points. Using the parameters of the procedure specified above, we can calculate error rates at different stages of the procedure. The results for data generated without noise and for the noisy data are represented in the figures below. Thereby, noisy data is generated in a way that 10% of image pixels revealed every trial are intentionally misclassified. Altogether, we produce for each of the cases three figures which represent the following:

1. Estimated posterior expectation in original Bubbles after 1, 2, 3 and 5 updates (i.e. 100, 200, 300 and 500 trials) where 100 trials are used between each update.
2. Estimated posterior expectation in adaptive Bubbles with Ising prior after 1, 2, 3 and 5 updates. 100 iterations are used between each update and the number of MCMC iterations accounts to 1000.
3. Error rates after different numbers of trials comparing the *original Bubbles* and the *original Bubbles with Bayesian adaption* (upper panel) and comparing the *original Bubbles* and the *adaptive Bubbles with Ising prior* (lower panel).

Overall, we can observe that the adaptive Bubbles with Ising prior outperforms the original Bubbles and the original Bubbles with Bayesian adaption in a sense of estimated error rates. While the original Bubbles with Bayesian adaption procedure performs only slightly better than the original Bubbles, adaptive Bubbles with Ising prior provides significantly better results. After the first update, we observe nearly the same misclassification rate for all three procedures. Further, error rates for original Bubbles remain nearly the same since the procedure allows to convert classified 'black' pixels to 'white' and vice versa, those for the original Bubbles with Bayesian adaption decrease steadily due to removing already classified pixels. The adaptive Bubbles with Ising prior performs best leading to a rapid decrease of error rates after the second update, i.e. after the MCMC has been run at least once and 'important' sampling regions have been identified. Updating the information a couple of times we apparently exploit more and more pixels from 'important' sampling regions and thus, approach towards the situation when proposed regions for further sampling contain more 'background' rather than 'object' pixels. In this situation we might observe a slight increase in error rates on later stages of the procedure.

Data without noise: original Bubbles

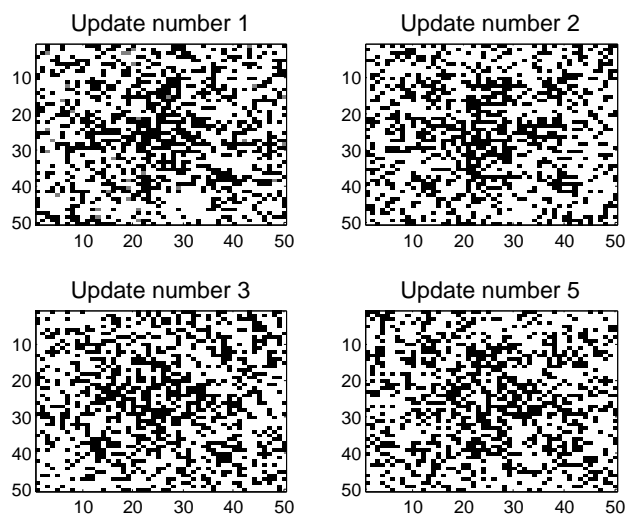


Fig. 4.2: Estimated posterior expectation in original Bubbles after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels are revealed.

Data without noise: adaptive Bubbles with Ising prior

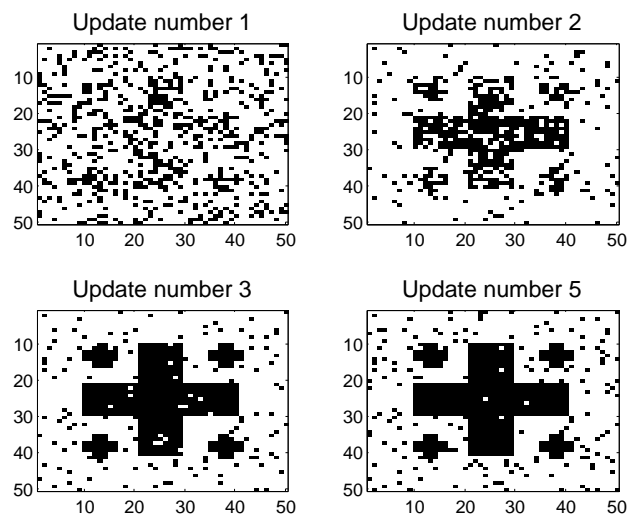


Fig. 4.3: Estimated posterior expectation in adaptive Bubbles with Ising prior after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels is revealed. Number of MCMC iterations is 1000.

Data without noise:
original Bubbles vs. adaptive Bubbles with Ising prior

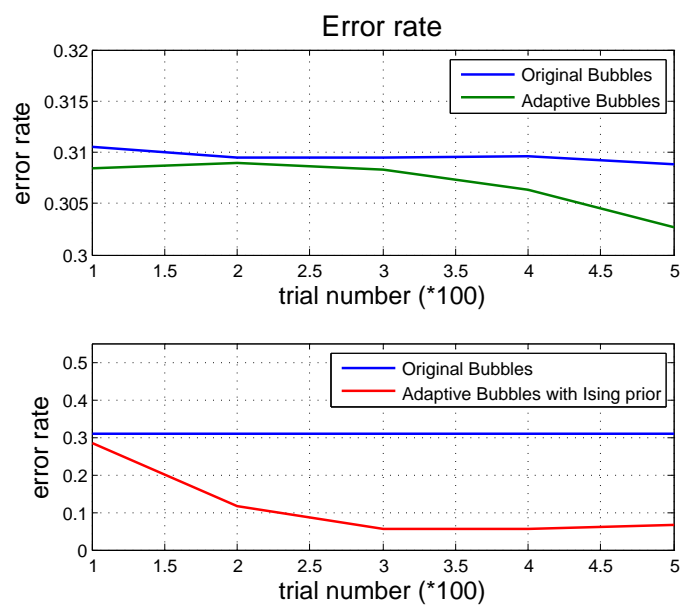


Fig. 4.4: Error rates after different numbers of trials: original Bubbles vs. original Bubbles with Bayesian adaption (upper panel) and original Bubbles vs. adaptive Bubbles with Ising prior (lower panel).

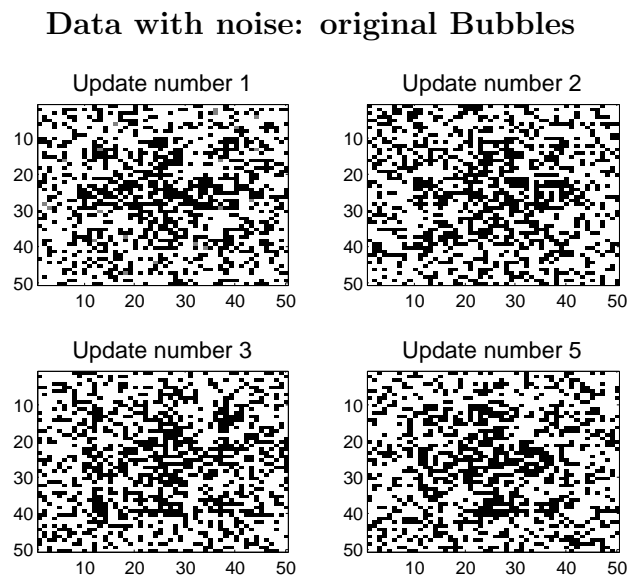


Fig. 4.5: Estimated posterior expectation in the original Bubbles approach after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels are revealed.

Data with noise: adaptive Bubbles with Ising prior

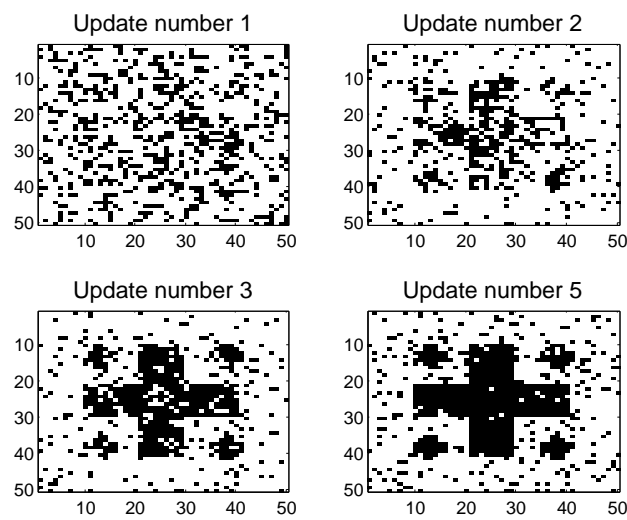


Fig. 4.6: Estimated posterior expectation in adaptive Bubbles with Ising prior after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels are revealed. Number of MCMC iterations is 1000.

Data with noise:
original Bubbles vs. adaptive Bubbles with Ising prior

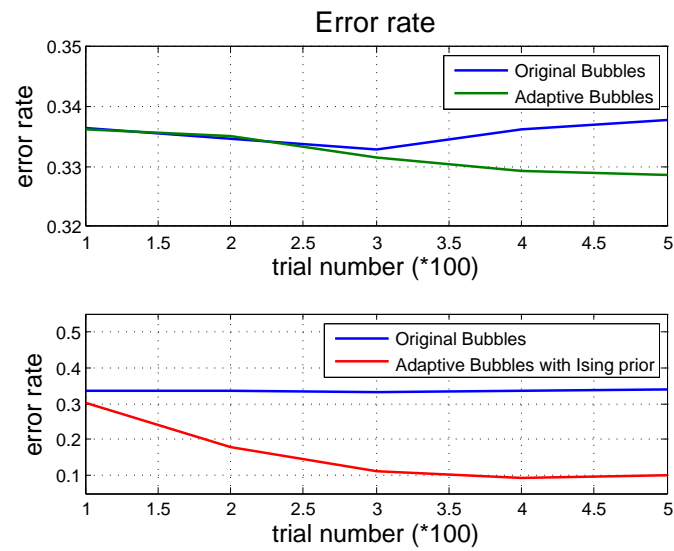


Fig. 4.7: Error rates after different numbers of trials: original Bubbles vs. original Bubbles with Bayesian adaption (upper panel) and original Bubbles vs. adaptive Bubbles with Ising prior (lower panel).

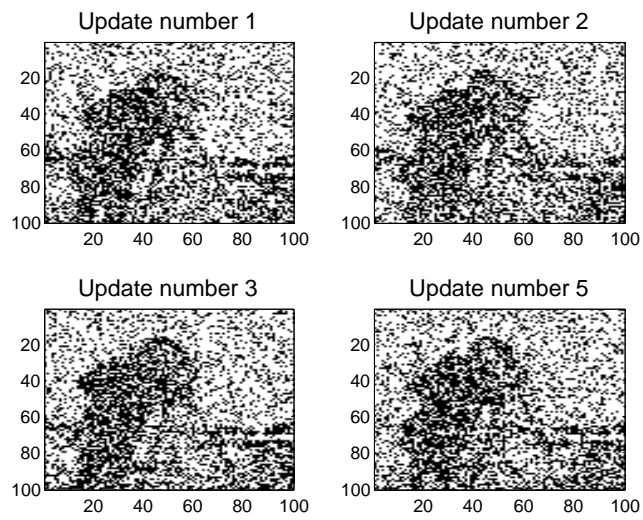
Data without noise: original Bubbles

Fig. 4.8: Estimated posterior expectation in original Bubbles approach after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels is revealed.

Data without noise: adaptive Bubbles with Ising prior

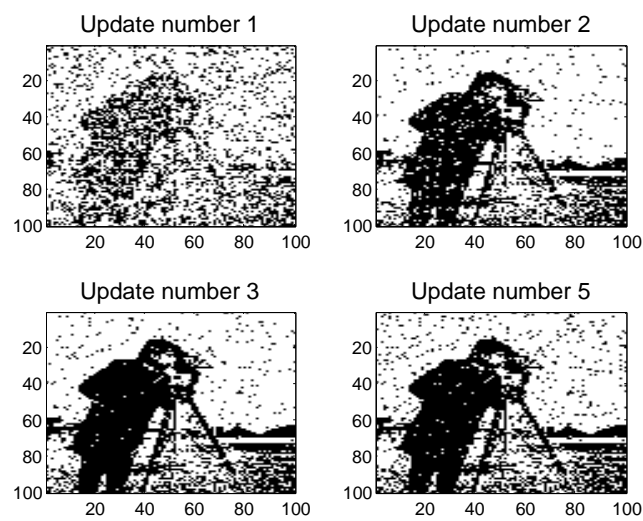


Fig. 4.9: Estimated posterior expectation in adaptive Bubbles with Ising prior after 1, 2, 3 and 5 updates with 100 trials between each update. In each trial 10% of image pixels is revealed. Number of MCMC iterations is 1000.

Data without noise:
original Bubbles vs. adaptive Bubbles with Ising prior

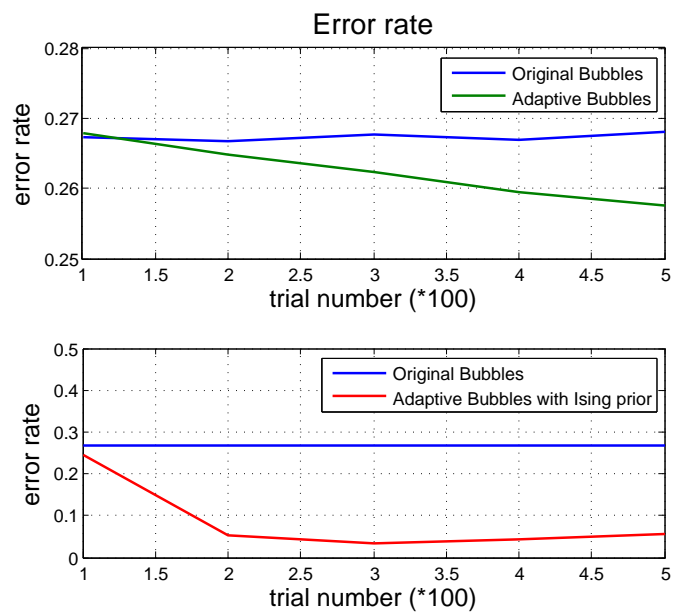


Fig. 4.10: Error rates after different numbers of trials: original Bubbles vs. original Bubbles with Bayesian adaption (upper panel) and original Bubbles vs. adaptive Bubbles with Ising prior (lower panel).

5. APPLICATIONS TO REAL DATA EXAMPLES

In previous chapters we compared the performance of the adaptive Bubbles with Beta MRF/Ising prior and the original Bubbles with or without Bayesian adaption using simulated data examples: we generated hidden binary data and sampled in every trial a certain portion of image pixels, replicating thereby a Bubbles situation. Behavior of observer was simulated by assigning labels 'important' or 'unimportant' to each of the pixels revealed in a trial. The aim of this chapter is to apply the adaptive Bubbles with Beta MRF/Ising prior to real data problems, where the true data is a facial image allowing a binary response, e.g. represented face can be characterized as neutral or happy or as male or female. In the following real data examples, a human observer is present. After revealing partial image information in a trial, the observer has to classify this image information according to EXNEX: neutral or happy or GENDER: male or female.



Fig. 5.1: Faces used in experiment 1.

5.1 Experiment 1: EXNEX

In this experiment we aim to determine which regions of input information are used by the observer to classify a facial image as neutral or happy. We run the experiment using 20 facial images of 10 different identities (5 males and 5 females), each displaying either a neutral or a happy expression, see Figure 5.1. All images are of size 100×100 . In every trial, human observer is presented with a partially revealed facial image, selected randomly from one of the facial images in Figure 5.1. The observer is asked to classify this partially revealed information according to its expression: neutral or happy. For revealing faces partially, we create a so-called *Bubble mask* - mid-grey

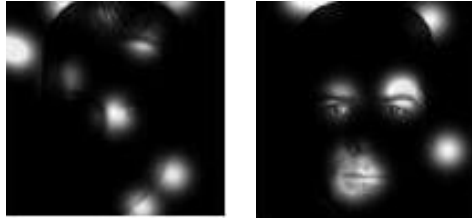


Fig. 5.2: Bubble mask in original Bubbles (left panel) and adaptive Bubbles with Beta MRF/Ising prior (right panel).

mask punctured by a number of Gaussian windows (the number of windows is set to 10), located either randomly (original Bubbles, left panel of Figure 5.2) or concentrated at 'important' regions (adaptive Bubbles with Beta MRF/Ising prior, right panel of Figure 5.2). Each Gaussian window represents a circle window with a center at the Bubble and a standard deviation (which is set equal to 7), controlling dispersion of the Gaussian. After revealing image partially, observer is asked to classify presented information as neutral or happy. If the response is correct (i.e. it coincides with an expression observer was presented with), we add the corresponding Bubble mask to the *CorrectPlane*. Thus, all Bubble masks leading to correct classification of a particular expression will be summed up to the *CorrectPlane*. The *TotalPlane* is a sum of all Bubble masks (both, leading to correct or incorrect classification). Then, as in simulated experiments, after every trial we compute the observed ratio r_i for every image pixel i in the following way:

$$r_i = \frac{n_i}{N_i}, \quad (5.1)$$

where n_i and N_i correspond to the pixel i^{th} value in *CorrectPlane* and in *TotalPlane*, respectively. Estimated posterior probability maps for the original Bubbles and the original Bubbles with Bayesian adaption after different numbers of trials are represented in the first and the second column of Figure 5.3, respectively. From the figure we can observe that the input information region used by the observer to classify a face as neutral or happy, is a mouth region. In order to obtain the posterior probability map for the adaptive Bubbles with Beta MRF/Ising prior, we run 1000 MCMC iterations every 100 trials. A third and a fourth column of Figure 5.3 represent the results for the adaptive Bubbles with Beta MRF prior and the adaptive Bubbles with Ising prior, respectively. Thereby, as in experiments with a simulated data, we exclude from further sampling those pixels which have posterior probability values higher than 0.95 or lower than 0.05. Further, weights are assigned to all remaining pixels, which will be sampled in the following trials. Weighting function calculated for non-excluded pixels after the first update (i.e. after running 1000 MCMC iterations for the first time) is represented in Figure 5.4. In order to compare the performance of the original Bubbles with or without Bayesian adaption and the adaptive Bubbles with Beta MRF/Ising

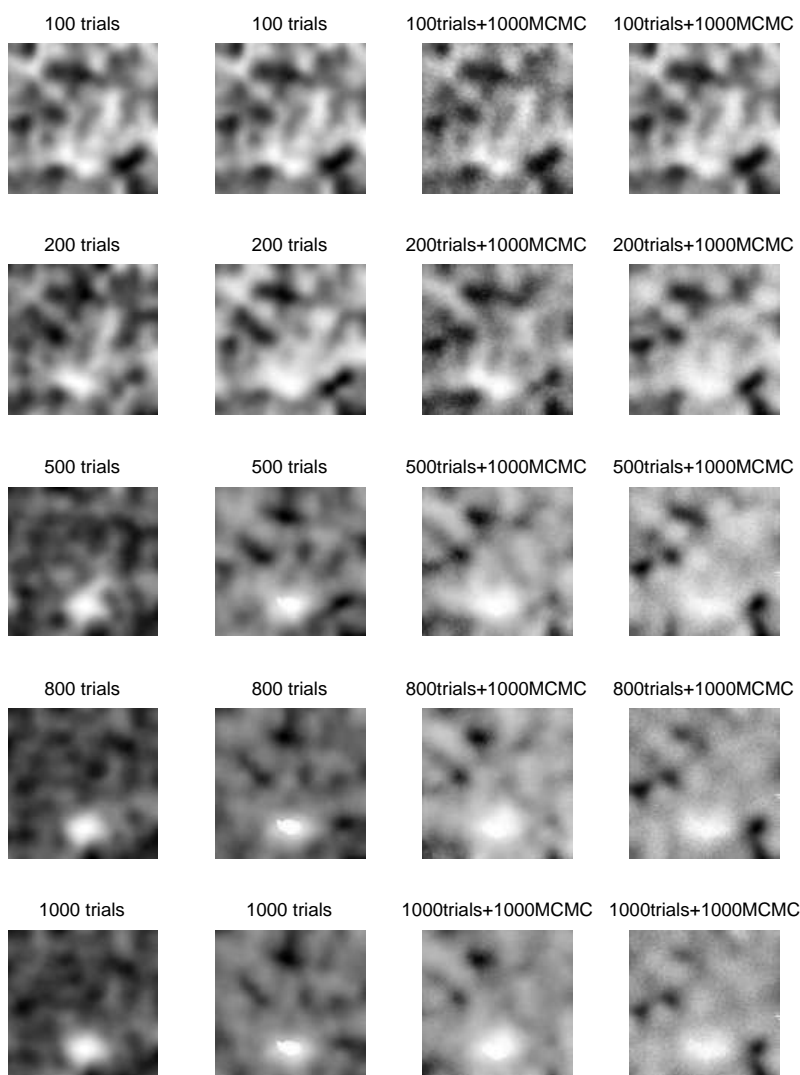


Fig. 5.3: Estimated posterior probability map for the original Bubbles (first column), original Bubbles with adaptation (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column) after different numbers of trials. The number of MCMC iterations run between each update (every 100 trials) for the adaptive Bubbles with Beta MRF/Ising prior is 1000.

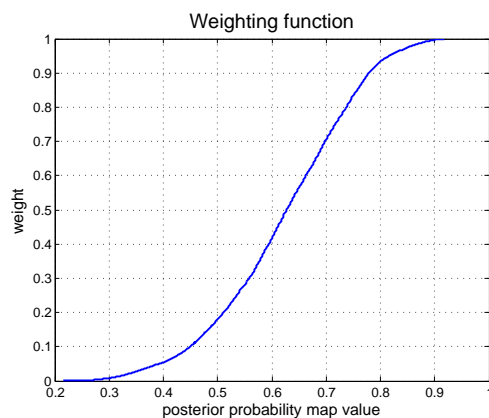


Fig. 5.4: Weighting function computed using posterior probability values after the first update (i.e. after running 1000 MCMC iterations for the first time). Image pixels will be sampled with this weights in the following trials.

prior, we calculate the *actual performance* given by the proportion of times observer has been correct with a particular expression to the total number of times this expression has been presented. These proportions for 'neutral' and 'happy' are presented in the upper and the lower panel of Figure 5.5, respectively. We can observe that already after the first update (i.e. after running 1000 MCMC for the first time), proportion of correctly classified expressions for the adaptive Bubbles with Beta MRF/Ising prior increases sharply. After it reaches its highest value corresponding to approximately 80% for both, neutral and happy expressions, it remains nearly constant across further trials. For the original Bubbles with Bayesian adaption the increase in correct

classified expressions can be explained by introducing a weighting scheme for sampling pixels from 'important' image regions (those with higher posterior probability map values) rather than sampling pixels randomly as proposed in original Bubbles. For the original Bubbles, the correct classification ratio corresponds to approximately 70%.

In order to determine, how input information regions evolve across trials, we produce posterior probability map images after performing *z-scoring* at 90% level, see Figure 5.6. Z-scored posterior probabilities are obtained by standardizing (subtracting mean and dividing by a standard deviation) of posterior probability map values from Figure 5.3 and then smoothing values greater than 1.65 (which is a 90% quantile of Normal distribution). For all four procedures we observe clusters of pixels with high posterior probability values concentrated around the mouth region.

In the following, we are interested in how the information growth gained through increasing the number of trials, affects the actual performance. If the number of pixels revealed and excluded from further sampling (Figure 5.7) does not change significantly across trials, we can stop sampling due to insignificant improvement in correct classification ratio. In order to find this *optimal stopping rule*, we compare posterior probability maps after different numbers of trials by measuring their closeness to each other (the distance between two posterior probability maps). This is done by computing the

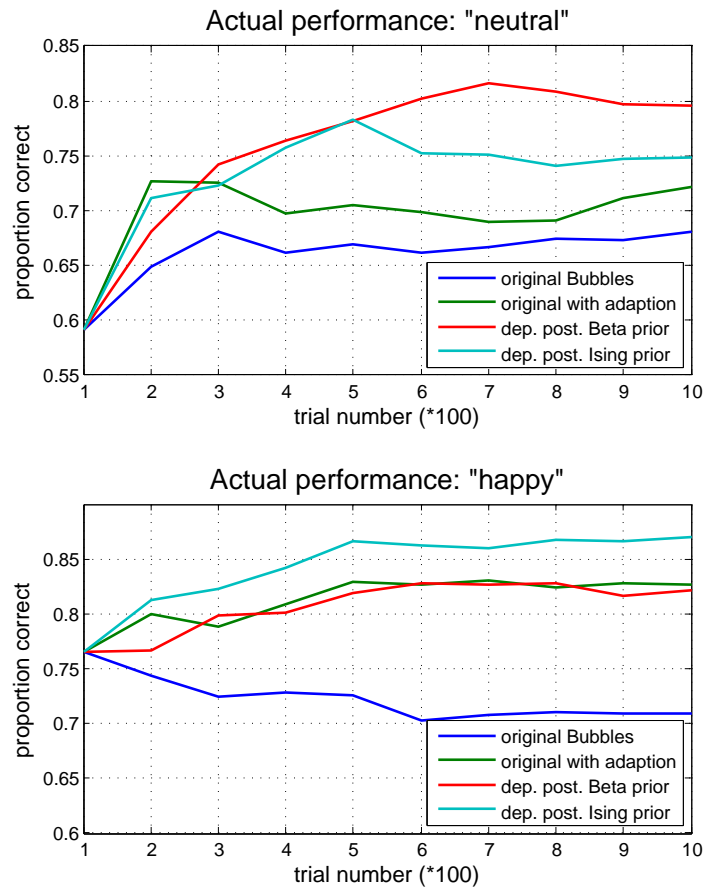


Fig. 5.5: Actual performance: proportion of times observer has been correct with a particular expression to the total number of times this expression has been presented. For 'neutral' (upper panel) and 'happy' (lower panel).

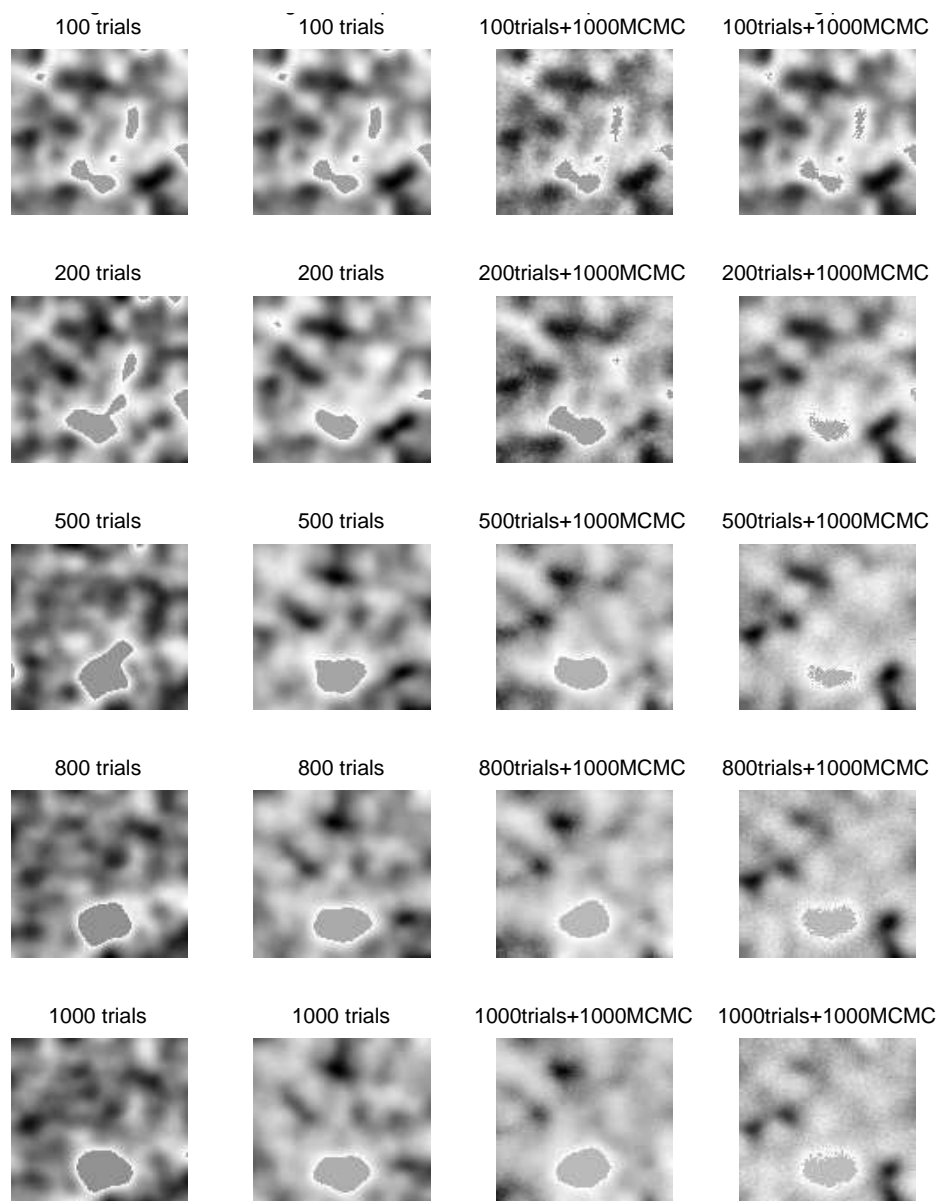


Fig. 5.6: Estimated posterior probability map after performing z-scoring at 90%-level to the estimates in Figure 5.3 for the original Bubbles (first column), original Bubbles with Bayesian adaptation (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column).

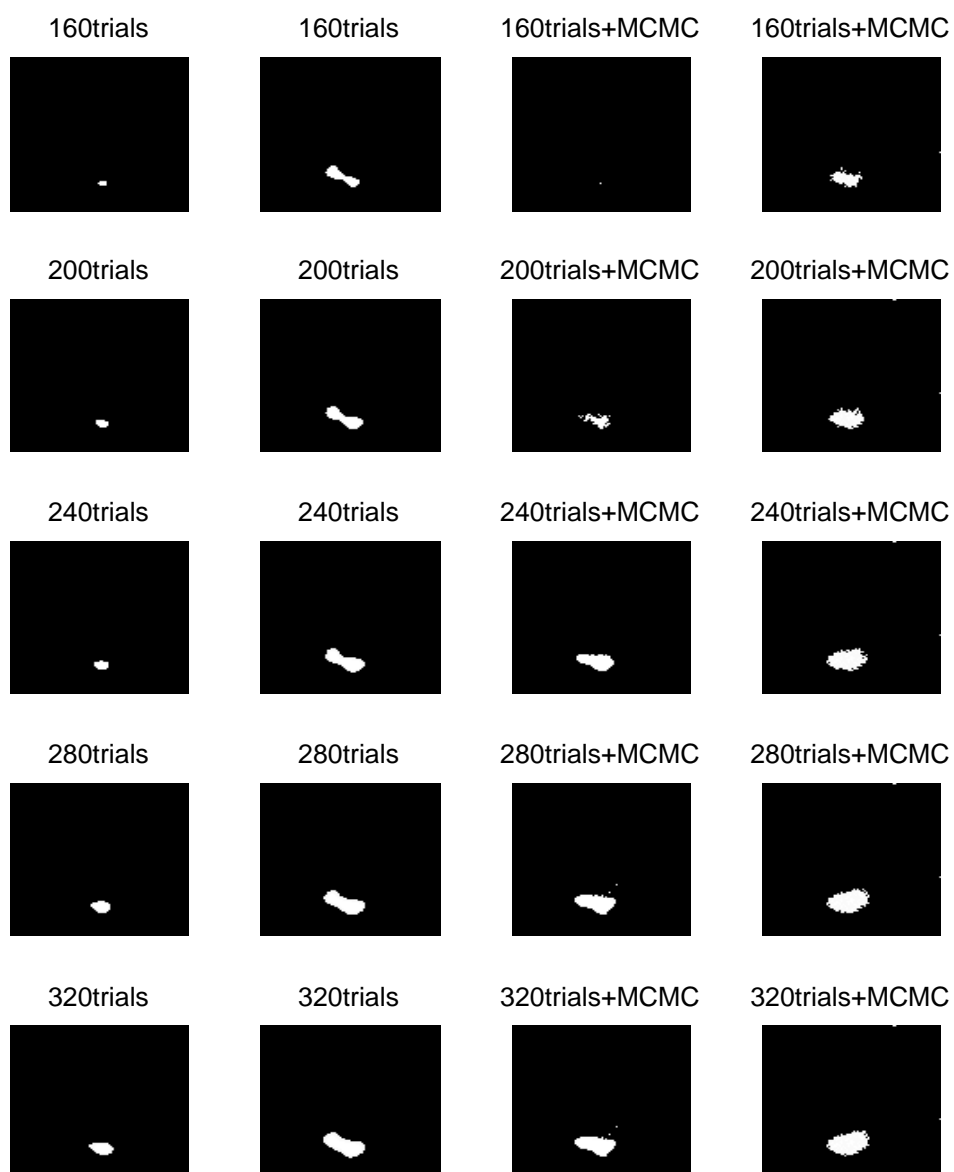


Fig. 5.7: Pixels with posterior probability values of 0.95 and higher for the original Bubbles (first column), original Bubbles with Bayesian adaptation (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column).

Euclidean distance for two posterior probability maps x^j and x^{j+k} after the j^{th} or the $(j+k)^{\text{th}}$ trial, respectively. It is given by the *Euclidean norm* $\|x^j - x^{j+k}\|_2$:

$$\rho(x^j, x^{j+k}) = \|x^j - x^{j+k}\|_2 = \sqrt{\sum_{i=1}^n (x_i^j - x_i^{j+k})^2}. \quad (5.2)$$

The smaller $\rho(x^j, x^{j+k})$ is, the closer are x^j and x^{j+k} . In the following, we run the experiment updating every 10 trials, i.e. we run 1000 MCMC iterations after 10, 20, 30, ... trials. The upper panel of Figure 5.8 shows how many pixels with posterior probability map values of 0.95 and higher have been revealed after different numbers of trials. The lower panel of Figure 5.8 shows the Euclidean distance after trial t , computed as $\rho(x^t, x^{400})$ - a distance between the posterior probability map after trial t and a posterior probability map after trial 400 (we do not continue sampling further due to insignificant changes in posterior probability map values). From the figure we can observe that the number of revealed and excluded pixels remains nearly constant after a certain number of trials (approx. 280 for Beta MRF prior and 320 for Ising prior) and the distance (similarity measure) goes to zero. Thus, implementing the adaptive Bubbles with Beta MRF prior, we can stop sampling after 280 trials, whereas in case of the Ising prior we stop sampling after approximately 320 trials.

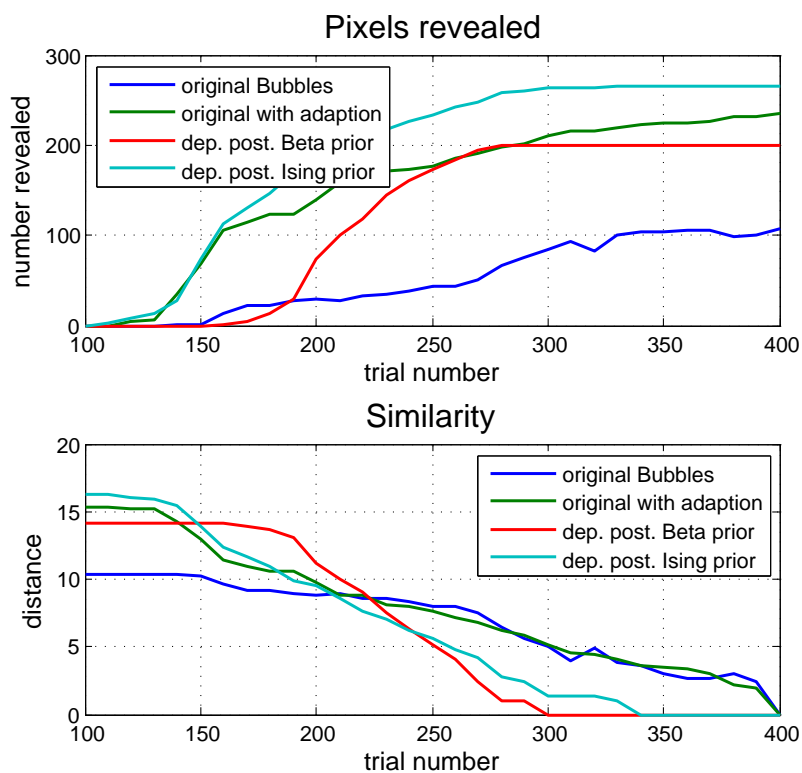


Fig. 5.8: Optimal stopping rule: how many pixels with high posterior probability values have been revealed after different number of trials (upper panel) and the Euclidean distance (lower panel) after different number of trials.

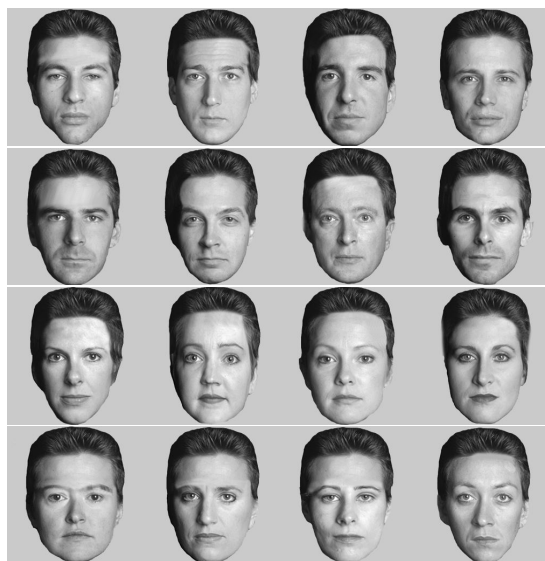


Fig. 5.9: Faces used in experiment 2.

5.2 Experiment 2: GENDER

In this experiment we are interested in which regions of input information are used by the observer to classify facial image information according to GENDER: male or female. We run the experiment using 16 facial images of different identities (8 males and 8 females), see Figure 5.9. All images are of size 100×100 pixels. In every trial we partially reveal one of the images in Figure 5.9 and ask the observer to classify a displayed information as male or female. Similar to the first experiment, Gaussian windows are positioned either randomly (original Bubbles, left panel of Figure 5.10), or at 'important' regions (adaptive Bubbles with Beta MRF/Ising prior, right panel of

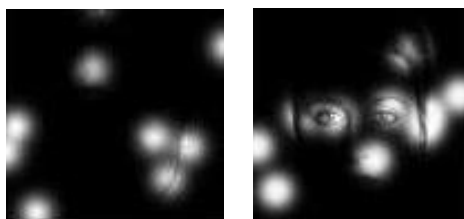


Fig. 5.10: Bubble mask in the original Bubbles (left panel) and the adaptive Bubbles with Beta MRF/Ising prior (right panel).

Figure 5.10). Estimated posterior probability maps for the original Bubbles, original Bubbles with Bayesian adaption, adaptive Bubbles with Beta MRF prior and adaptive Bubbles with Ising prior are represented in Figure 5.11. As in the previous experiment, the number of MCMC iterations accounts to 1000 between every 100 trials. From the figure we can observe that the input information region used by the observer to classify facial information as male or female, is an eye and a mouth region. In the following, we compare the *actual performance* of original Bubbles with or without adaption and the adaptive Bubbles Beta MRF/Ising prior. Figure 5.12 represents the proportion of times observer has been correct with a particular expression to the total number of times this expression has been presented for male (upper panel) and female (lower panel). We can observe that adaptive procedures allow to reach values for correct classifications higher than the original Bubbles: there is a sharp increase in correctly classified expressions after the first

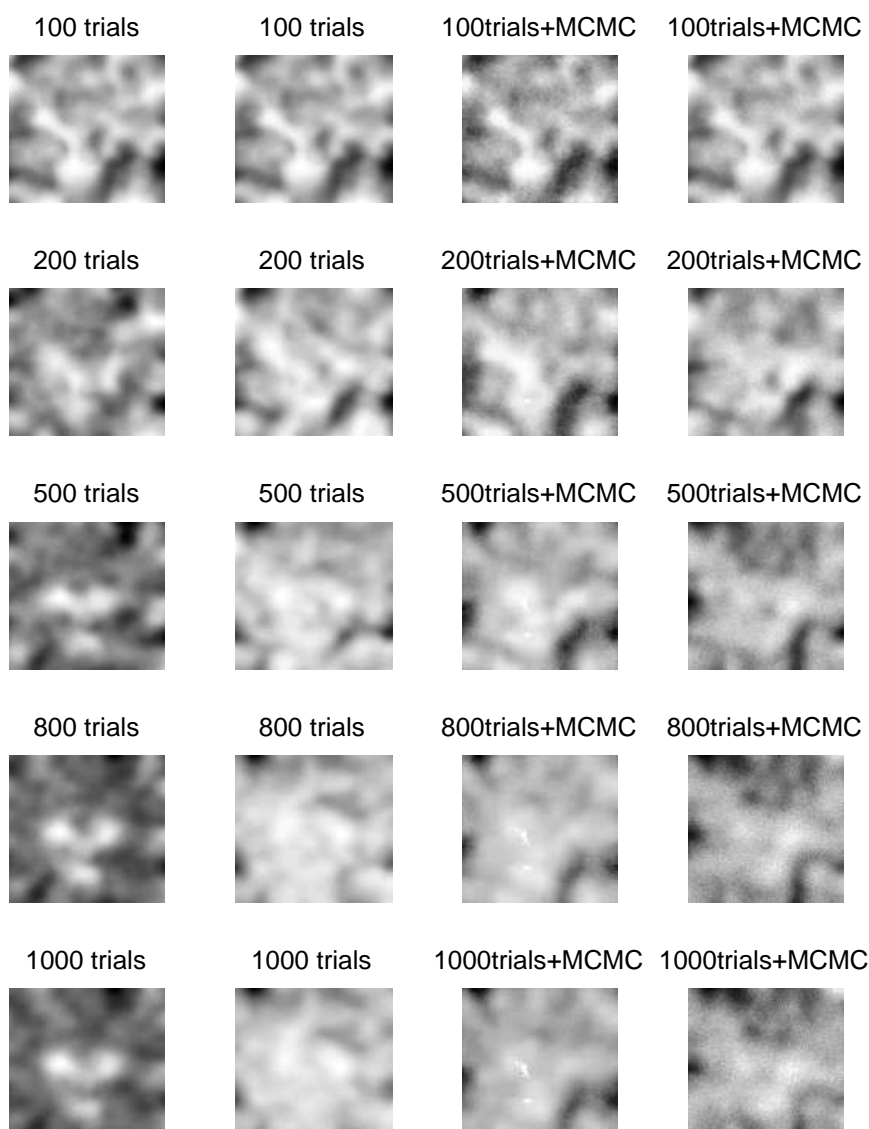


Fig. 5.11: Estimated posterior probability map for the original Bubbles (first column), original Bubbles with Bayesian adaptation (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column) after different numbers of trials. The number of MCMC iterations run between each update (every 100 trials) for the adaptive Bubbles with Beta MRF/Ising prior is 1000.

100 trials and further minor increase across trials. Correct classification ratio for the original Bubbles remains nearly constant across trials due to the random sampling of image pixels. Figure 5.13 represents posterior probability map images after performing *z-scoring* at 90% level. Here again, we observe clusters of pixels with high posterior probability values, concentrated around the eye and the mouth region.

As in the first example, we are interested in choosing a stopping rule which determines when sampling ends. For this purpose we observe how the number of pixels with high posterior probability map values evolves across trials and we stop sampling if it does not change significantly. Therefore, we run the experiment updating every 10 trials, i.e. we run 1000 MCMC iterations after 10, 20, 30, ... trials. The upper panel of Figure 5.15 shows the number of pixels with 'high' posterior probability values after different numbers of trials. In adaptive procedures, these pixels will be revealed and excluded from further sampling. Therefore, the value 0.9 has been chosen for the original Bubbles and the adaptive Bubbles with Beta MRF prior. For the original Bubbles with adaptation and the adaptive Bubbles with Ising prior, we choose the value 0.95 since lower values would reveal pixels from 'unimportant' image regions. From the figure we can observe, that the adaptive Bubbles with Beta MRF prior provides more pronounced results without revealing pixels from the background regions. The *Euclidean distance* for measuring similar-

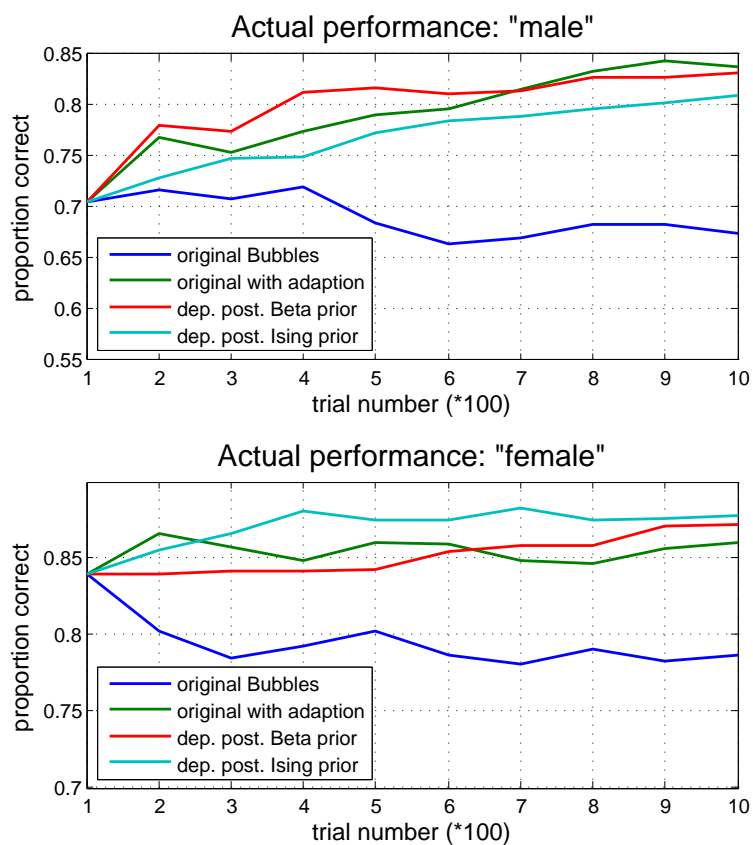


Fig. 5.12: Actual performance: proportion of times observer has been correct with a particular expression to the total number of times this expression has been presented. For 'male' (upper panel) and 'female' (lower panel).

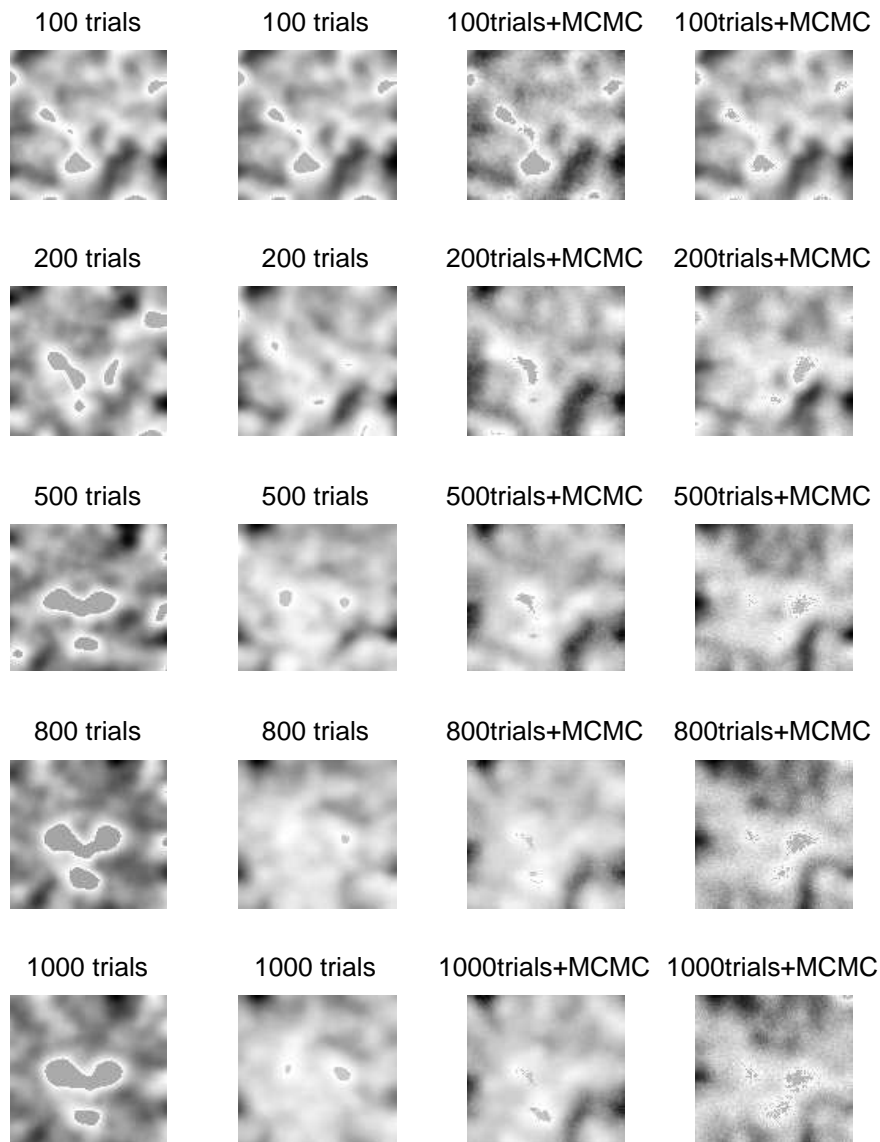


Fig. 5.13: Estimated posterior probability map after performing z-scoring at 90%-level to the estimates in Figure 5.11 for the original Bubbles (first column), original Bubbles with adaptation (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column).

ity between two posterior probability maps x^j and x^{620} is plotted in the lower panel of Figure 5.15. While the original Bubbles does not allow to decide on the optimal stopping rule due to random pixel sampling, we observe that the optimal number of trials for stopping sampling in all adaptive procedures accounts to approximately 450.

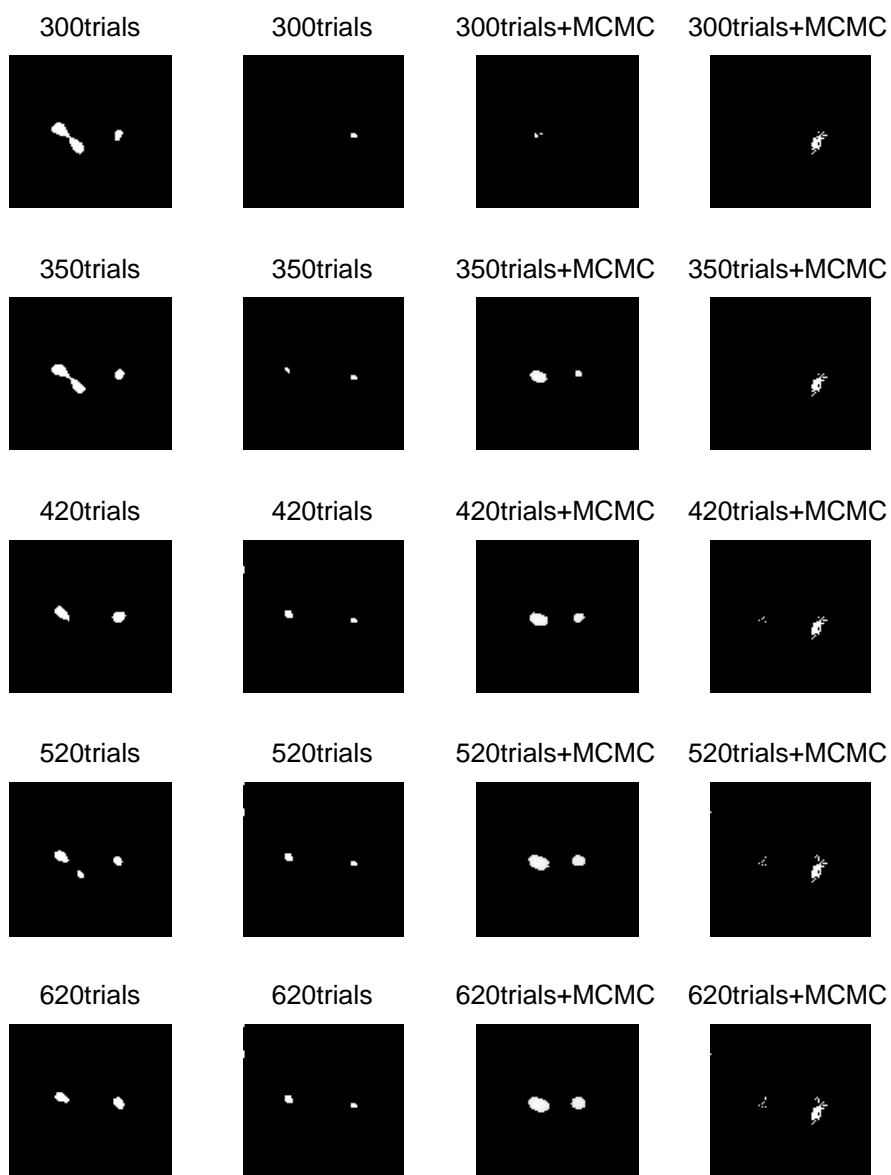


Fig. 5.14: Pixels with high posterior probability values for the original Bubbles (first column), original Bubbles with adaptation (second column), adaptive Bubbles with Beta MRF prior (third column), adaptive Bubbles with Ising prior (fourth column).

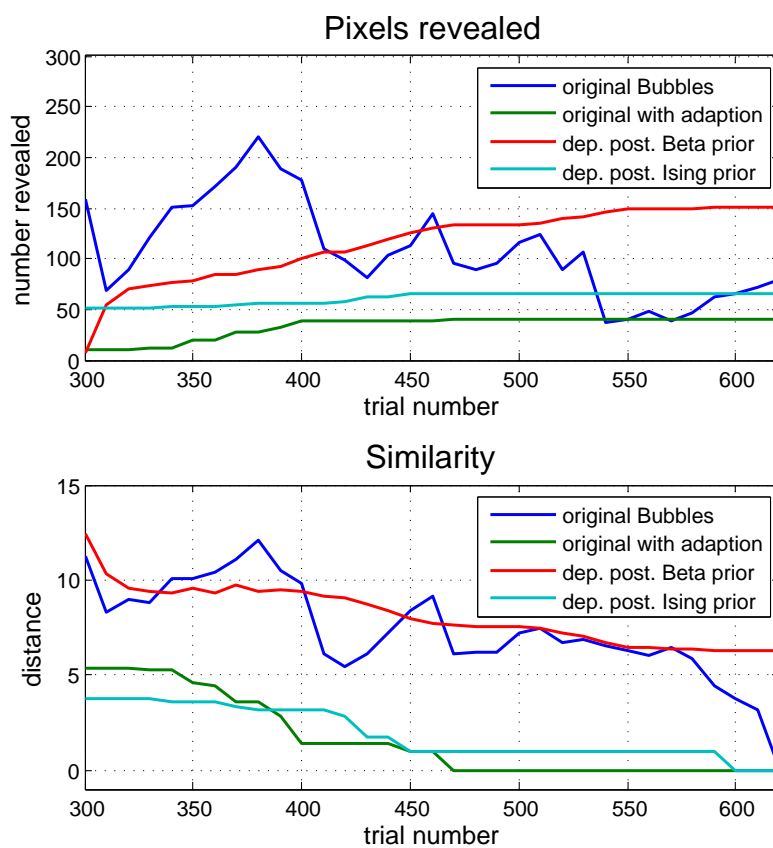


Fig. 5.15: Optimal stopping rule: how many pixels with high posterior probability values have been revealed after different number of trials (upper panel) and the Euclidean distance (lower panel) after different number of trials.

6. INFERENCE FOR HYPERPARAMETERS

In this chapter we aim to make inference about all unknown model parameters applied to the situation where the prior is a Beta MRF, i.e. we consider the adaptive Bubbles with Beta MRF prior (we can clearly apply the below consideration to the case where the prior is given by the Ising model). In case of the Beta MRF prior, the unknown model parameters include the latent probabilities p as well as the vector of parameters $(\alpha, \beta, \theta)^\top$. The full posterior distribution for all unknown parameters is given by

$$\pi(p, \alpha, \beta, \theta | y^1, \dots, y^j) \propto L(y^1, \dots, y^j | p) \cdot \pi(p | \alpha, \beta, \theta) \cdot \pi(\alpha, \beta, \theta). \quad (6.1)$$

This posterior distribution could be sampled using a Metropolis within Gibbs algorithm where all parameters are sampled from their full conditional distributions. Now, the full-conditional distribution for p appears as

$$\pi(p | y^1, \dots, y^j, \alpha, \beta, \theta) \propto L(y^1, \dots, y^j | p) \cdot \pi(p | \alpha, \beta, \theta) \quad (6.2)$$

which presents no difficulties to compute and is identical to the algorithm described in chapter 3 for fixed $(\alpha, \beta, \theta)^\top$ values.

However, the MH update for $(\alpha, \beta, \theta)^\top$ would consist in sampling from

$$\pi(\alpha, \beta, \theta | y^1, \dots, y^j, p) \propto \pi(p | \alpha, \beta, \theta) \cdot \pi(\alpha, \beta, \theta). \quad (6.3)$$

Suppose we propose to move from $(\alpha, \beta, \theta)^\top$ to $(\alpha^*, \beta^*, \theta^*)^\top$. Then, the acceptance probability would be

$$\min \left\{ 1, \frac{\pi(p | \alpha^*, \beta^*, \theta^*) \cdot \pi(\alpha^*, \beta^*, \theta^*)}{\pi(p | \alpha, \beta, \theta) \cdot \pi(\alpha, \beta, \theta)} \right\}, \quad (6.4)$$

which requires computation of the joint distribution $\pi(p | \alpha, \beta, \theta)$. However, in practice, it is only known up to normalizing constant, which itself depends on (α, β, θ) . One approach to overcome this problem consists in approximation of $\pi(p | \alpha, \beta, \theta)$ by the *pseudo-likelihood estimator* $\pi^*(p | \alpha, \beta, \theta)$, which represents the product of full conditional probabilities $\pi(p_i | p_{-i}, \alpha, \beta, \theta)$ for each lattice point $i = 1, \dots, n$:

$$\pi^*(p | \alpha, \beta, \theta) \approx \prod_{i=1}^n \pi(p_i | p_{\setminus i}, \alpha, \beta, \theta). \quad (6.5)$$

(Ryden & Titterton 1998) applied the above algorithm in case where the hidden process is an autologistic distribution.

In the following, we aim to explore the performance of the pseudo-likelihood estimator in estimating parameter values from realized Beta MRF distributions .

6.1 Parameter Inference in Beta MRFs

In this section we are moving away from applying the pseudo-likelihood estimator in the scheme described above where the interest is in sampling from $\pi(p, \alpha, \beta, \theta | y^1, \dots, y^j)$ to the simpler scenario of using the pseudo-likelihood estimator to maximize $\pi(p | \alpha, \beta, \theta)$. Using a previous Beta MRF set-up, we will estimate the unknown parameters $(\alpha, \beta, \theta)^\top$ from the full conditional Beta-density

$$\pi(p_i | p_{-i}) \propto p_i^{\{\alpha - \sum_{k \sim i} \theta \cdot \log(1 - p_k)\}} \cdot (1 - p_i)^{\{\beta - \sum_{k \sim i} \theta \cdot \log(p_k)\}} \cdot \frac{\Gamma(A_{i,1} + 1) \Gamma(A_{i,2} + 1)}{\Gamma(A_{i,1} + A_{i,2} + 2)} \quad (6.6)$$

with

$$\begin{aligned} A_{i,1} &= \alpha - \sum_{k \sim i} \theta \cdot \log(1 - p_k) \text{ and} \\ A_{i,2} &= \beta - \sum_{k \sim i} \theta \cdot \log(p_k), \end{aligned}$$

by maximizing the likelihood $L\{\pi(p | \alpha, \beta, \theta)\}$. As before, $i \sim k$ means here that pixel i is a neighbor of pixel k . Likelihood calculation requires the knowledge of the joint distribution $\pi(p | \alpha, \beta, \theta)$ and since it is unknown in practice, we approximate the likelihood function by the *pseudo-likelihood function* (Besag 1974), substituting (6.6) into (6.5):

$$L\{\pi^*(p | \alpha, \beta, \theta)\} = \prod_{i=1}^n p_i^{\{\alpha - \sum_{k \sim i} \theta \cdot \log(1 - p_k)\}} \cdot (1 - p_i)^{\{\beta - \sum_{k \sim i} \theta \cdot \log(p_k)\}} \cdot \frac{\Gamma(A_{i,1} + 1) \Gamma(A_{i,2} + 1)}{\Gamma(A_{i,1} + A_{i,2} + 2)}. \quad (6.7)$$

Alternatively to maximization of (6.7), taking the logarithm on both sides of equation (6.7), we come up with maximization of the *pseudo log-likelihood*:

$$\begin{aligned}
 l\{\pi^*(p|\alpha, \beta, \theta)\} &= \left\{ \alpha - \sum_{k \sim i} \theta \cdot \log(1 - p_k) \right\} \cdot \log(p_i) \\
 &+ \left\{ \beta - \sum_{k \sim i} \theta \cdot \log(p_k) \right\} \cdot \log(1 - p_i) \\
 &+ \log \{\Gamma(A_{i,1} + 1)\} + \log \{\Gamma(A_{i,2} + 1)\} - \log \{\Gamma(A_{i,1} + A_{i,2} + 2)\}.
 \end{aligned} \tag{6.8}$$

Thus, the estimates of a vector of unknown parameters is obtained by

$$(\hat{\alpha}, \hat{\beta}, \hat{\theta})^\top = \operatorname{argmax}_{\alpha, \beta, \theta} l\{\pi^*(p|\alpha, \beta, \theta)\}. \tag{6.9}$$

This method has been employed in a wide variety of settings. In particular, it has been used in context of MRFs for example by (Ryden & Titterton 1998) and (Descombes, Morris, Zerubia & Berthod 1999) to estimate the vector of unknown parameters in different prior models.

6.2 Simulated Annealing Algorithm

Maximization of a pseudo log-likelihood function (6.8) requires a global optimization technique, such as the *simulated annealing (SA) algorithm* introduced in (Kirkpatrick, Gelatt & Vecchi 1983). The name of the algorithm comes from annealing of metals, glass or liquids. In the process of annealing, a material is heated and then cooled, usually for softening and making the material less brittle. Therefore, the algorithm exposes the initial solution

(initial position of atoms) to heat (maximizing initial energy) and cools afterwards (minimizing of initial energy), providing a more optimal solution.

SA algorithm samples iteratively from

$$\pi_m^*(p|\alpha, \beta, \theta) \propto \{\pi^*(p|\alpha, \beta, \theta)\}^{1/T_m}, \quad (6.10)$$

where, as above, $\pi^*(p|\alpha, \beta, \theta) = \prod_{i=1}^n \pi(p_i|p_{-i}, \alpha, \beta, \theta)$ and T_m is a temperature parameter. When T_m goes to 0 ($m \rightarrow \infty$) slowly enough, the algorithm generates a Markov chain which converges in distribution towards the uniform distribution over the set of configurations maximizing $\pi^*(p|\alpha, \beta, \theta)$. To implement the algorithm, we first have to define the sampling method and a cooling schedule.

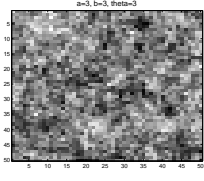
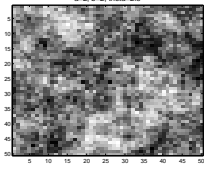
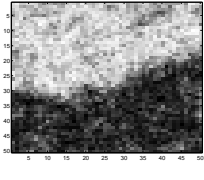
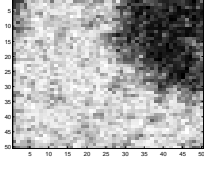
The cooling schedule of the SA algorithm consists in reducing the temperature from initially set high value T gradually in every following simulation step towards $T = 0$. One possible choice of a cooling schedule is exponential, where the temperature decreases by a fixed factor $0 < \tau < 1$ at each following simulation step.

The sampling is implemented by means of the MCMC algorithm, in each step of the SA algorithm, the current solution is replaced by a random solution which is close to the current one. Therefore, all 'better' solutions (leading to the higher likelihood) are always accepted with probability 1. However, if the solution is 'worse' (i.e. the value of the likelihood is lower than the pre-

vious likelihood value), it still can be accepted with a *transition probability* depending on a difference between the old and the new likelihood values and on the temperature decreasing during the process. Thus, the *transition* or *acceptance* probability of a new state, can be defined in a following way:

$$p_{acc} = \begin{cases} 1 & \text{if } l_m^* > l_{m-1}^* \\ \exp \left\{ \frac{l_m^* - l_{m-1}^*}{T \cdot \tau^m} \right\} & \text{otherwise} \end{cases}$$

Using the SA algorithm, we estimate parameter values for different realizations of Beta MRFs. Therefore, we run for every temperature value 1000 MCMC iterations, proposing new random 'nearby' values of parameters. The initial temperature is set to $T = 100$ and the decreasing factor is $\tau = 0.5$. Table 6.1 represent estimated parameter results. We can observe that the closer is parameter θ to 0, the more precise are the estimation results. This evidence is not surprising due to the assumption that lattice points are independent, which is achieved when parameter $\theta = 0$.

True parameter values	MRF realization	Estimated parameter values
$\alpha = 3.0, \beta = 3.0, \theta = 3.0$		$\hat{\alpha} = 2.9325, \hat{\beta} = 2.9491, \hat{\theta} = 2.9043$
$\alpha = 2.0, \beta = 2.0, \theta = 2.5$		$\hat{\alpha} = 2.1057, \hat{\beta} = 2.1154, \hat{\theta} = 2.4752$
$\alpha = 1.5, \beta = 1.5, \theta = 2.5$		$\hat{\alpha} = 1.5383, \hat{\beta} = 1.5445, \hat{\theta} = 2.5211$
$\alpha = 1.0, \beta = 1.0, \theta = 2.0$		$\hat{\alpha} = 0.9836, \hat{\beta} = 0.9759, \hat{\theta} = 1.9708$

Tab. 6.1: Pseudo-likelihood estimation for Beta MRFs parameters α , β and θ .

Estimated using simulated annealing algorithm.

7. SUMMARY AND FURTHER RESEARCH

The main objective of my thesis was to improve the *original Bubbles approach* represented in (Gosselin & Schyns 2001). For this purposes, three alternative methods have been presented: the *original Bubbles with Bayesian adaption*, the *adaptive Bubbles with Beta MRF prior* and the *adaptive Bubbles with Ising prior*. All three approaches address the problem of ineffective sampling in *original Bubbles* by introducing the excluding rule for pixels with 'very high' ($p \geq 0.95$) or 'very low' ($p \leq 0.05$) posterior probability map values and applying a weighed sampling scheme to all remaining pixels, i.e. those pixels which have posterior probability map values in the range (0.05, 0.95). Clearly, the sampling scheme gives 0 weight to pixels outside this range, and thus, behaves discontinuously at $p = 0.05$ and $p = 0.95$. As a further improvement one could consider some alternative weighting schemes, e.g. ones continuous in the posterior probability, that might perform even better. Overall, the adaptive methods implemented in this way allow to reduce the number of sampling trials making the procedure less exhaustive.

Incorporating spacial dependence in the image is important, not necessarily to reduce trials further, but to achieve better estimation results. Here we considered two different ways of implementation: the *adaptive Bubbles with Beta MRF prior* and the *adaptive Bubbles with Ising prior*. The *adaptive Bubbles with Beta MRF prior* assumes that the hidden values of p take values in interval $[0, 1]$ which correspond to the probability that the pixel is important. Thus, Beta MRF was used as a prior distribution to make inference for unknown probabilities p . This method can be regarded as an extension of the *original Bubbles with Bayesian adaption* where $Beta(1, 1)$ distribution is used as a prior, but where image pixels are assumed to be independent. An alternative approach, the *adaptive Bubbles with Ising prior* uses the *autologistic model* as a prior distribution and thus, suggests to make inference about the unknown binary values $x \in \{-1, +1\}$, which correspond to whether pixel is 'important' or not. Comparing these two methods in terms of the number of sampling trials, we observed, that the Beta MRF prior allows us to stop sampling slightly earlier than the Ising prior.

The way in which both algorithms have been implemented assumes that the vector of parameters $(\alpha, \beta, \theta)^\top$ for the Beta MRF prior or $(\alpha, \beta)^\top$ for the Ising prior are fixed throughout our experiment. Further work can be done to incorporate these in our Bayesian model: instead of sampling from the fixed parameter model $\pi(p|y^1, \dots, y^j, \alpha, \beta, \theta)$, we can sample from the distribution

$\pi(p, \alpha, \beta, \theta | y^1, \dots, y^j)$ using the pseudo-likelihood method discussed in chapter 6 for its approximation. Adjusting parameter values on-line, we expect to achieve better estimation results.

BIBLIOGRAPHY

- Besag, J. (1974), ‘Spatial interaction and the statistical analysis of lattice systems’, *Journal of the Royal Statistical Society* **B**(36), 192–236.
- Darroch, J. N., Lauritzen, S. L. & Speed, T. (1980), ‘Markov fields and log-linear interaction models for contingency tables’, *The Annals of Statistics* **8**(3), 552–539.
- Descombes, X., Morris, R., Zerubia, J. & Berthod, M. (1999), ‘Estimation of markov random field prior parameters using markov chain monte carlo maximum likelihood’, *IEEE Transactions on Image Processing* **8**(7), 954–963.
- Geman, S. & Geman, D. (1984), ‘Stochastic relaxation, gibbs distributions, and the bayesian restoration of images’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741.
- Gosselin, F. & Schyns, P. (2001), ‘Bubbles: a technique to reveal the use of information in recognition tasks’, *Vision research* **41**(17), 2261–2271.

-
- Hastings, W. (1970), ‘Monte Carlo Sampling Methods Using Markov Chains and Their Applications’, *Biometrika* **57**(1), 97–109.
- Kaiser, M. & Cressie, N. (2000), ‘The construction of multivariate distributions from markov random fields’, *Journal of Multivariate Analysis* (73), 199–200.
- Kirkpatrick, S., Gelatt, C. & Vecchi, M. (1983), ‘Optimization by simulated annealing’, *Science* **220**(4598), 671–680.
- Metropolis, N., Rosenbluth, A., Rosenbluth, R., Teller, A. & Teller, E. (1953), ‘Equation of State Calculations by Fast Computing Machines’, *Journal of Chemical Physics* **21**, 1087–1091.
- Pettitt, A., Friel, N. & Reeves, R. (2003), ‘Efficient calculation of the normalizing constant of the autologistic and related models on the cylinder and lattice’, *J.R. Statist. Soc. B* **65**, Part 1, 235–246.
- Rue, H. & Held, L. (2005), *Gaussian Markov Random Fields*, Chapman & Hall London.
- Ryden, T. & Titterton, D. (1998), ‘Computational bayesian analysis of hidden markov models’, *Journal of Computational and Graphical Statistics* **7**(2), 194–211.