Segmentation of Textured Satellite and Aerial Images by Bayesian Inference and Markov Random Fields

Simon P. Wilson, Josiane Zerubia

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Segmentation of Textured Satellite and Aerial Images by Bayesian Inference and Markov Random Fields.

Simon P. Wilson — Josiane Zerubia

N° 4336

December 2001

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--4336--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Segmentation of Textured Satellite and Aerial Images by Bayesian Inference and Markov Random Fields.

Simon P. Wilson , Josiane Zerubia

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Ariana

Rapport de recherche n° 4336 — December 2001 — 25 pages

Abstract: We investigate Bayesian solutions to image segmentation based on the double Markov random field model, originally proposed by Melas and Wilson. Inference on the number of classes in the image is done via reversible jump Metropolis moves. These moves, usually implemented by splitting and merging classes, can be very slow, making them impractical for large images. We investigate simpler reversible jump moves that are quick to implement but show that they may mix very slowly. We propose a more complex split and merge scheme and compare its performance. Tests are conducted on satellite and aerial images.

Key-words: unsupervised image segmentation, Markov chain Monte Carlo, Markov random field, reversible jump, satellite images, aerial images

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Segmentation d'images satellitaires et aériennes texturées par inférence bayésienne et champs de Markov

Résumé : Nous étudions un modèle markovien double, initialement proposé par Melas et Wilson, pour la segmentation d'image. Le nombre de classes de l'image est obtenu par inférence bayésienne via un algorithme de Metropolis à saut réversible. Les mouvements habituellement utilisés dans une telle dynamique consistent en la fission ou la fusion de classes. Mais cela peut nécessiter beaucoup de temps de calcul, en particulier sur des images de grande taille. Ici, nous étudions des mouvements plus simples qui sont rapides à mettre en oeuvre, mais dont la mélangeance peut être longue. Nous proposons alors un schéma de fission/fusion plus complexe et comparons les performances obtenues. Nous effectuons des tests sur des images satellitaires et aériennes.

Mots-clés : segmentation d'image non-supervisée, méthodes de Monte Carlo par chaîne de Markov, champs de Markov, saut réversible, images satellitaires, images aériennes

Contents

1 Introduction

A double Markov random field is a hierarchical model for an image composed of several textures (or classes), where both the grey levels of each class and the class labels are described by Markov random fields. It has been extensively used in Bayesian approaches to image segmentation, see [10] for instance.

In this report, we investigate the use of such models for unsupervised image segmentation, where both the number of texture classes and their properties are unknown. We simulate from an approximation to the posterior distribution of labels and parameters by Markov chain Monte Carlo methods. To handle the unknown number of classes, reversible jump moves are proposed, at first based on adding or deleting empty classes. Our goal is to investigate whether these simple reversible jump moves, which have the advantage of being very quick to implement, are in fact enough to efficiently explore the space of texture classes. Later we consider more complicated moves based on splitting and merging non-empty classes. Our conclusion is that these more complicated moves are necessary to analyse all but the simplest of images.

Reversible jump methods [4] have been proposed for image segmentation in [1] and [6]; in the former case, a double Markov random field model was proposed, in the latter case colour images were modelled, but each pixel colour was independent given the labels. In both these approaches, moves between different numbers of classes were made by splitting or merging existing classes. In the case of [1], which uses a double Markov random field, the split move is computationally very burdensome, in order to give a reasonable acceptance probability of the move. Although the method seems quite successful, this makes it impractical for large images, say of size 3000×3000 like one encounters in satellite imaging. Our reversible jump split and merge moves attempt to be simpler and involve less computation, allowing such large images to be processed.

The paper is organised as follows. Section 2 describes the double Markov random field. Section 3 demonstrates algorithms for segmentation that use the double Markov random field when the number of classes is known, and illustrates the approach with examples. In Section 4, the idea of reversible jump moves for sampling the number of classes is introduced, and the algorithm of Section 3 is extended to include this step. Examples are given on satellite and aerial images. Finally, concluding remarks are given in Section 5.

2 The Double Markov Random Field

Consider a rectangular lattice of pixel sites $\mathcal{S}$. An image consists of an array of grey values $(x_s)_{s\in\mathcal{S}}$ and labels $(y_s)_{s\in\mathcal{S}}$, identifying the texture type present. Let there be R textures in the image and each texture, defined on all of $\mathcal{S}$, is a Markov random field T^r, parameterised by θ_r, with neighbourhood system having set of cliques $\mathcal{C}_r$. The label process is another Markov random field Y, parameterised by β and with neighbourhood system having set of cliques $\mathcal{C}_Y$. All the fields are independent conditional on model parameters and their

distributions have the following Gibbs representation:

$$P(T^r = t \mid \theta_r) \;=\; \frac{\exp[-U_r(t;\theta_r)]}{Z_r(\theta_r)} \;=\; \frac{\exp\left[-\sum_{c\in\mathcal{C}_r} V_{r,c}(t;\theta_r)\right]}{Z_r(\theta_r)}, \tag{1}$$

$$P(Y = y \mid R, \beta) \;=\; \frac{\exp[-U_Y(y;R,\beta)]}{Z_Y(R,\beta)} \;=\; \frac{\exp\left[-\sum_{c\in\mathcal{C}_Y} V_{Y,c}(y;R,\beta)\right]}{Z_Y(R,\beta)}. \tag{2}$$

The partition functions Z_r and Z_Y cannot usually be evaluated.

Definition: The *double Markov random field* is a probability distribution over X and Y where the T^r and X are Markov random fields and the observed grey value at pixel s is $X_s = T_s^{Y_s}$. The joint distribution for X and Y is therefore:

$$P(X = x, Y = y \mid R, \theta_1, \ldots, \theta_R, \beta) \;=\; P(Y = y \mid R, \beta) \prod_{r=1}^{R} P(T^r_{S_r} = x_{S_r} \mid \theta_r), \tag{3}$$

where $S_r = \{s \in \mathcal{S} \mid y_s = r\}$ and $T^r_{S_r}$, x_{S_r} denote T^r and x restricted to S_r.

Usually, Y is assumed to be a Potts model and the X are modelled by Gaussian Markov random fields, although obviously other choices are available. In what follows we assume these models are being used unless otherwise stated.

2.1 Boundary Assumptions

In order to evaluate $P(T^r_{S_r} = x_{S_r} \mid \theta_r)$ it is necessary to identify the interaction structure of pixels that have missing neighbours. This is generally intractable, leading to the use of various boundary assumptions. The marginal distribution on the regions is then approximated by the joint distribution of the sites in S_r conditioned upon the assumed boundary values. One common boundary approximation is the free boundary, where pixels on the edges of the region have fewer neighbours than those in the interior. This modifies the conditional distribution of the pixels on the edges; re-scaling the clique functions at edge sites is a possible way of correcting this effect. Another possibility is a fixed boundary, where the missing neighbouring values of the pixels on the edges are replaced by arbitrary fixed values. The often used toroidal structure is not applicable in this case because the sets of pixels S_r are not necessarily rectangular. From Equation 1, under a free boundary structure, the probability functions for each region containing a single texture will be given by

$$P(T^r_{S_r} = x_{S_r} \mid \theta_r) \;=\; \frac{\exp[-\sum_{c\in\mathcal{C}_r \mid c\subset S_r} V_{r,c}(x_c \mid \theta_r)]}{Z_{S_r}(\theta_r)}. \tag{4}$$

Equations 2, 3 and 4 define more fully the double Markov random field, a phrase that was to our knowledge first used in [13].

2.2 Using the Model for Bayesian Segmentation

In a Bayesian approach to segmentation, the goal is to calculate the distribution of Y, and the model parameters if unknown, given X, that is:

$$P(Y, R, \theta_1, \ldots, \theta_R, \beta \mid X) \propto P(X, Y \mid R, \theta_1, \ldots, \theta_R, \beta)\, P(R, \theta_1, \ldots, \theta_R, \beta), \tag{5}$$

where $P(R, \theta_1, \ldots, \theta_R, \beta)$ is a prior distribution on the parameters. The only available technique to evaluate this posterior distribution is Monte Carlo Markov chain (MCMC) simulation, usually the Gibbs sampler, where one simulates from the "full conditionals" of each Y_s, $s \in \mathcal{S}$, and if necessary of β and θ_r, $r = 1, \ldots, R$. To simulate from R, other methods are needed, as described in Section 4. The parameters are usually assumed independent:

$$P(R, \theta_1, \ldots, \theta_R, \beta) = P(R)\, P(\beta) \prod_{r=1}^{R} P(\theta_r).$$

If Y is the Potts model then β is the inverse temperature and a uniform prior over [0,5] is assumed, which covers the critical value of the model and allows both weak and strong clustering of class labels. When R is assumed known, of course $P(R)$ is degenerate, and when unknown a proper prior must be assumed; geometric or Poisson are common choices. We always further assume that there is some finite maximum number of classes $R_{\max}$ so that the $P(R)$ is restricted to $1 \leq R \leq R_{\max}$. When textures are modelled by Gaussian Markov random fields, each θ_r consists of a mean μ_r, variance σ_r^2 and clique parameters $\theta_{r1}, \ldots, \theta_{rK}$, the number K depending on the neighbourhood system used. We assume a uniform prior distribution for μ_r over the range $[0, 255]$, an inverse gamma prior for σ_r^2 and a uniform prior for $(\theta_{r1}, \ldots, \theta_{rK})$ over the allowable range of values; see [7] for this range in the case of a second order neighbourhood system. An inverse gamma prior for σ_r^2 has the form $P(\sigma_r^2) \propto [\sigma_r^2]^{-(a+1)} \exp(-b/\sigma_r^2)$ for hyperparameters a and b, and is used because it is the conjugate prior for the Gaussian variance, making the sampling of this parameter easier (it is another inverse gamma), and also because this distribution can be specified with a thick tail, making it more robust (typically this is done by letting $a = b = 0.5$). Markov random field models are practical in image analysis because the full conditional distributions have a simple form, making MCMC algorithms like Gibbs sampling easy to implement. Unfortunately, for the double Markov random field, the full conditional distribution for the pixel labels is not of such an easy form. The Markov property is lost because the partition functions $Z_{S_r}(\theta_r)$ are functions of Y, and so $P(Y_s \mid X, R, \beta, \theta_1, \ldots, \theta_R, Y_{-s})$ is not only a function of neighbouring class labels. In addition, if the parameters are unknown, their full conditional distributions are usually not available either; this is true for the Potts–Gaussian model that we use here.

So, another modification to the model must be made, this time to recover the Markov property of the full conditional of Y_s. In [10], it was shown that many Bayesian approaches to segmentation can be viewed as double Markov random field models, with different modifications made in order to make $P(Y_s \mid X, R, \beta, \theta_1, \ldots, \theta_R, Y_{-s})$ available for simulation. In an extensive simulation study of 5 such modifications, the following *pseudo-likelihood approach* performed best.

2.3 The Pseudo-likelihood Approach

In this approach, we approximate $P(X, Y \mid R, \beta, \theta_1, \ldots, \theta_R)$ by the product of full conditionals:

$$P(X, Y \mid R, \beta, \theta_1, \ldots, \theta_R) \;\approx\; P(Y \mid R, \beta) \prod_{s \in \mathcal{S}} P(X_s \mid \theta_{Y_s}, X_t; t \in \eta_s); \tag{6}$$

where η_s is the neighbourhood of s. For each s, it is assumed that the texture Y_s holds in the entire neighbourhood of s. This approximation is developed in [2] and is called the pseudo-likelihood. This approximation allows us to simulate easily from the full conditionals $P(Y_s \mid X, R, \beta, \theta_1, \ldots, \theta_R, Y_{-s})$. Also, this approximation is sufficient to allow the full conditionals of β and the θ_r to be available in most cases, including the Potts–Gaussian model, and allows reversible jump moves for R to be implemented.

3 Segmentation with R Known

We can use Gibbs sampling to do image segmentation when R is known. The prior distributions described in the last section are assumed, with $P(R)$ degenerate. An initial random labelling is assumed for Y. Then in turn we sample from $P(Y_s \mid X, R, \beta, \theta_1, \ldots, \theta_R, Y_{-s})$ for each s, $P(\beta \mid R, Y)$ and $P(\theta_r \mid X, Y)$, all using the pseudo-likelihood approximation. In the case of the Potts-Gaussian model, the full conditional of Y_s is:

$$P(y_s = r | x, y_j, j \neq s, \theta_r, \beta) \propto$$
$$\frac{\exp\left[-\frac{1}{2\sigma_r^2}\left\{x_s - \mu_r - \sum_{k=1}^{K}\sum_{j|<s,j>_k} \theta_{rk}(x_j - \mu_r)\right\}^2\right]}{\sqrt{2\pi\sigma_r^2}} \times \exp\left[\beta \sum_{j \in \eta_s} \delta(r - y_j)\right], \tag{7}$$

where η_s is the neighbourhood of s and $< s, j >_k$ refers to the pixels s and j being of clique type k. Details of this approach may be found in [8] or [9]. After a burn-in period, samples are being drawn from the pseudo-likelihood approximation to the posterior distribution of $(Y, \beta, \theta_1, \ldots, \theta_R)$. Following the burn in, one can look at the most sampled label at each site. This is the MPM (Maximum Posterior Mode) segmentation:

$$\left(\arg\max_r P(Y_s = r \mid X)\right)_{s \in \mathcal{S}}.$$

If the MPM solution is sought, estimates of other quantities of interest can be calculated by this method. For example, the entropy in marginal posterior distribution of each label indicates the level of uncertainty in that label, given by:

$$e_s = -\sum_{\substack{j=1 \\ \hat{p}_{sj} > 0}}^{R} \hat{p}_{sj} \log(\hat{p}_{sj}),$$

where $\hat{p}_{sj}$ is the proportion of times y_s was sampled as class j. Also, by looking at the relative proportion of values sampled, estimates of posterior distributions of parameters can be made.

Alternatively, one can look for the MAP segmentation

$$\arg\max_y P(Y = y \mid X),$$

which requires the maximum of the posterior density to be found. This is done by Gibbs sampling in conjunction with simulated annealing. In simulated annealing, at the nth iteration, the algorithm samples the Y_s, β and θ_r from the distributions proportional to $P(Y_s \mid X, R, \beta, \theta_1, \ldots, \theta_R, Y_{-s})^{1/T_n}$, $P(\beta \mid R, Y)^{1/T_n}$ and $P(\theta_r \mid X, Y)^{1/T_n}$ for a *temperature* parameter $T_n > 0$. The sequence of T_n decreases to 0. Examples of such sequences are geometric ($T_n = T_0\rho^n$, for $0 < \rho < 1$) and logarithmic ($T_n = T_0/[1 + \log(n)]$). Theoretical results show that, for certain temperature sequences, the sampled values of Y, β and θ_r converge to the MAP. Unfortunately, sequences that satisfy the conditions of the theory converge very slowly near 0, and so in practice other faster decreasing sequences are used, or the sampler is stopped when the temperature is still some distance from 0, with the result that the solution found is usually only a local maximum of $P(Y = y \mid X)$. However, our experience is that such local maxima usually give a good segmentation. This method was first used for image analysis in [3], and details for segmentation with this approach are also in [8] or [9].

Some examples of segmentation with this algorithm are given in Figures 1, 2 and 3. Figure 1 shows a SPOT satellite image of an agricultural region of Holland, and Figure 2 shows results from applying this procedure. The MPM segmentation is computed, which also gives a map of the entropy, with blue indicating low entropy through green, yellow and to red indicating the highest entropy. This gives a measure of uncertainty in the class of each pixel, and we see that class 2 (having the second lowest mean value and coloured black in the segmentation) has the lowest uncertainty in general, and the highest uncertainty occurs at the borders between regions. The figure also shows an estimate of the posterior distribution of the mean of class 2. Similar plots can be made for all other model parameters, although we note that the pseudo-likelihood may not be a good approximation for the posterior distribution of clique parameters. Having recorded at each iteration the number of pixels in each class, we can construct the lower right plot, being an estimate of the posterior distribution of the percentage of pixels that are in this class, something that might be of interest in applications to land use estimation. Figure 3 shows MPM segmentation of an aerial image in the top left into 4, 6 and 8 classes.

The number of iterations used for these MPM segmentations was 5000, with the first 3000 ignored as burn in. In our experience, this seems adequate for the size of image and number of classes in these two cases. However, the number of iterations is very dependent on the image, and these figures are not a substitute for monitoring convergence of parameters and class sizes. Plotting the number of pixels per class as a function of iteration, as used later in Figure 5, is a good way to monitor if the algorithm has converged.

Figure 1: Image of an agricultural area of Holland.

If R is not known, one can apply this algorithm for many different R, define a measure for the success of a segmentation, and then select the best segmentation according to the criterion. For example, if ground truth for similar images was known, this could be used to determine a best R. However, a more objective approach would be to select R on the basis of the data. In the next section we discuss how this might be done.

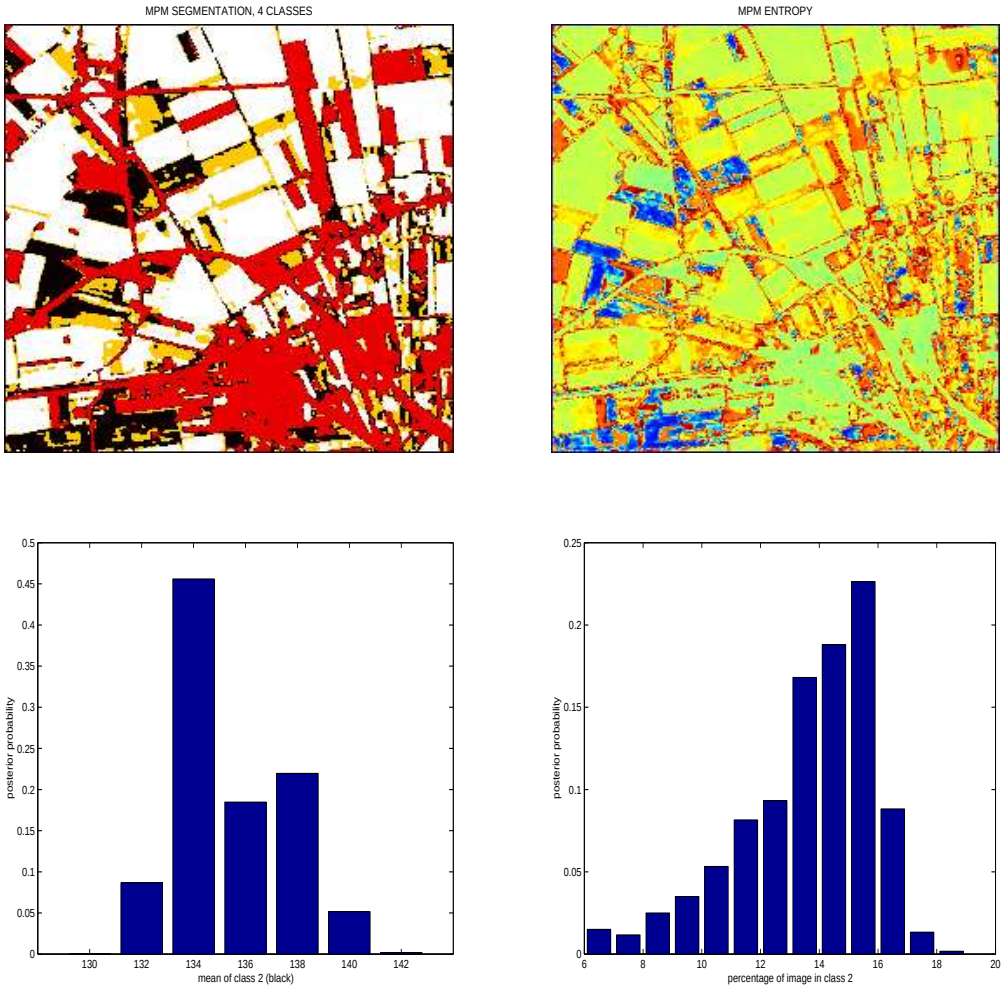

Figure 2: Analysis of the image in Figure 1 by the method with number of classes fixed at 4.

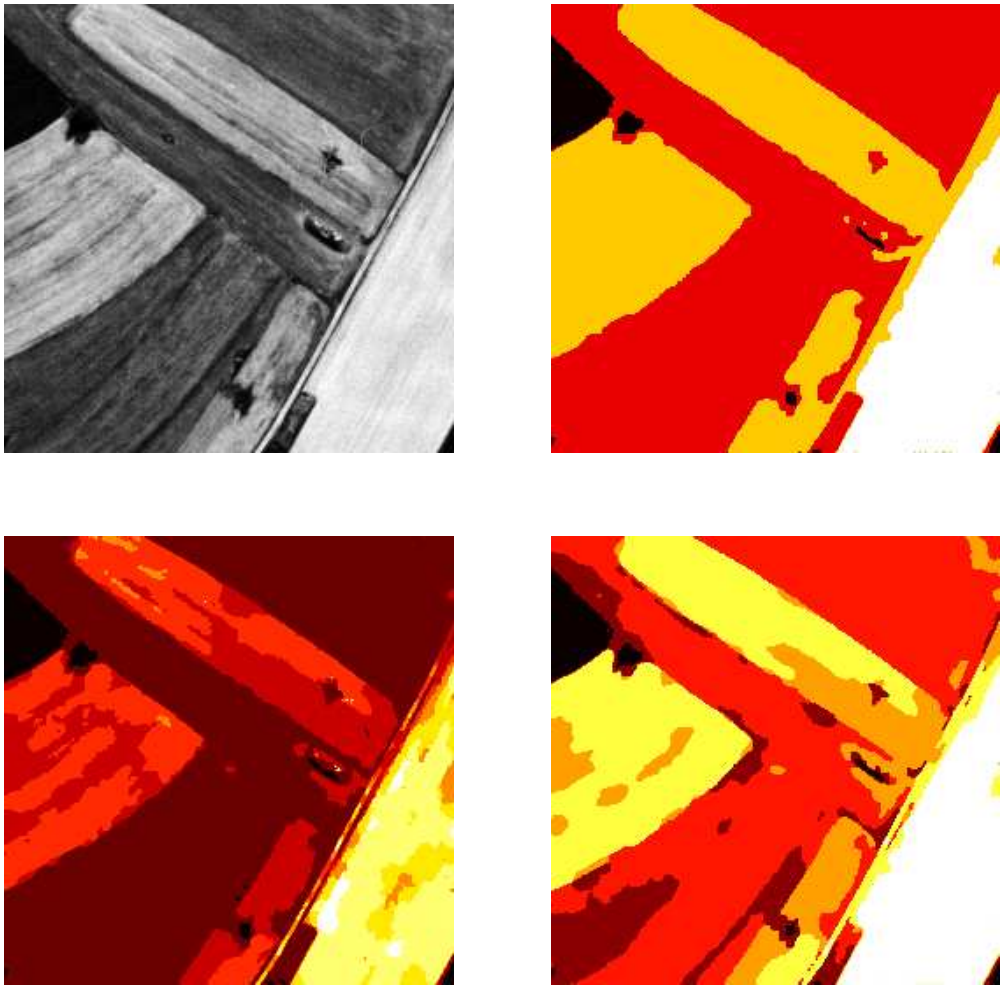

Figure 3: MPM segmentation of the image in the top left into (clockwise from top right) 4, 6 and 8 classes.

4 Segmentation with R Unknown

When R is unknown, the posterior distribution of Equation 5 now includes a proper prior $P(R)$ on the number of classes and our MCMC algorithm must be extended to include sampling from the full conditional distribution of R. This distribution cannot be explored by Gibbs sampling, since sampling different values of R will alter the dimension of the parameter space, by changing the number of texture parameters θ_r. Gibbs sampling will not work for spaces where the dimension changes in this fashion, but fortunately there are other Markov chain Monte Carlo sampling methods that can be used for R (see [12]). One of these is *reversible jump MCMC*.

Reversible jump has been used for identifying R in image segmentation by Markov random fields [1, 6]. In [6], segmentation was proposed for colour images, but the model used is not a double Markov random field, because pixel colours are assumed independent given the classes. In [1], the model proposed is a double Markov random field, and the pseudo-likelihood approximation is used. The move proposed to increase R is complicated, in order to ensure a reasonable acceptance probability. This renders that algorithm slow, and rather impractical for large images.

4.1 Reversible Jump Markov Chain Monte Carlo

Reversible jump was proposed in [4], which we refer to for a full description of the approach. The idea is to sample R by a Metropolis move. This requires an acceptance probability, and in this application the only moves that are both practical and may have a non-zero acceptance probability are to increase or decrease R by 1. Increasing R by 1 requires that we generate parameter values for a new class, and possibly a new segmentation that involves the new class. The new parameter values must be generated according to a 1-1 transformation between the new set of values and the old parameters together with random numbers necessary to generate the parameters for the new class. That is a move from $\Theta_R = (\theta_1, \ldots, \theta_R)$ to $\Theta^*_{R+1} = (\theta^*_1, \ldots, \theta^*_R, \theta^*_{R+1})$ is achieved with a set random numbers u of the same dimension as θ^*_{R+1} such that there is a 1-1 function

$$(\Theta_R, u) \leftrightarrow \Theta^*_{R+1}.$$

The usual conditions of reversibility and irreducibility must be maintained. Decreasing R by 1 requires that we eliminate one of the classes, and propose a segmentation where the deleted class is not present. The reversibility means that this move must be done using the same 1-1 function between Θ^*_{R+1} and (Θ_R, u). Once either move has been done, an acceptance probability can be computed. For the move to increase R to $R+1$, with a change in parameters from Θ_R to Θ^*_{R+1}, and a change in segmentation from Y to Y^*, this acceptance probability is the minimum of 1 and

$$\frac{P(X, Y^* \mid R+1, \Theta^*_{R+1}, \beta)}{P(X, Y \mid R, \Theta_R, \beta)} \frac{P(\Theta^*_{R+1})}{P(\Theta_R)} \frac{P(R+1)}{P(R)} \frac{1}{P(u)} \frac{P(\text{decrease } R+1)}{P(\text{increase } R)} \left| \frac{\partial(\Theta_R, u)}{\partial \Theta^*_{R+1}} \right|, \quad (8)$$

where the final partial derivative is the Jacobian of the transformation $(\Theta_R, u) \to \Theta^*_{R+1}$, and $P(\text{decrease } R+1)$ and $P(\text{increase } R)$ refer to any other probabilities involved in decreasing or increasing the number of classes (for example, randomly choosing which class to delete).

The above is a strategy for sampling for R. Sampling from the other parameter values and Y can then proceed by Gibbs sampling exactly as in the case of fixed R. This forms an MCMC algorithm whose stationary distribution is the pseudo-likelihood approximation of the posterior distribution in Equation 5.

4.2 Sampling R with Simple Reversible Jump Moves

Our goal is to find reversible jump moves that are not too computationally intensive. The simplest possible is to propose to add or delete classes that are "empty" in the sense that no pixel in the current segmentation belongs to that class. Because our old and new segmentations are identical, the ratio $P(X, Y^* \mid R+1, \Theta^*_{R+1}, \beta)/P(X, Y \mid R, \Theta_R, \beta)$ is unity so does not need to be calculated. Further, we generate new class parameters from the prior distribution $P(\theta)$. Thus u is generated from $P(\theta)$ and the 1-1 transformation $(\Theta_R, u) \leftrightarrow \Theta^*_{R+1}$ is the identity. The acceptance probability is then simply

$$\min\left\{1, \frac{P(R+1)}{P(R)} \frac{P(\text{decrease } R+1)}{P(\text{increase } R)}\right\}; \tag{9}$$

this can actually be calculated *prior* to generating the new class. The advantages of such an approach are much faster iterations of the sampler and considerably simpler code. The disadvantages are that we can only delete a class if it becomes empty, and therefore we will have slow mixing of the chain if this does not happen often, and that in proposing new texture parameters from the prior, we may only rarely propose texture classes that fit any portion of the image well, and so remain empty.

4.3 A Simple Segmentation Algorithm using Reversible Jump

We propose the following MCMC algorithm:

1. Initialise R. Decide on a number of iterations M. Randomly select starting values for Y, Θ_R and β as in the fixed R case.
2. Repeat M times:
   - Sample from Y, β and Θ_R as in the fixed R case.
   - If $R = R_{\max}$, decide to decrease R. If $R = 1$, decide to increase. Otherwise, with probability 0.5, decide to increase R else decrease:
     - If decide to increase, decide to accept with probability given in Equation 9. If accept, generate θ_{R+1} from the prior. Do not assign any of the labels to this class.

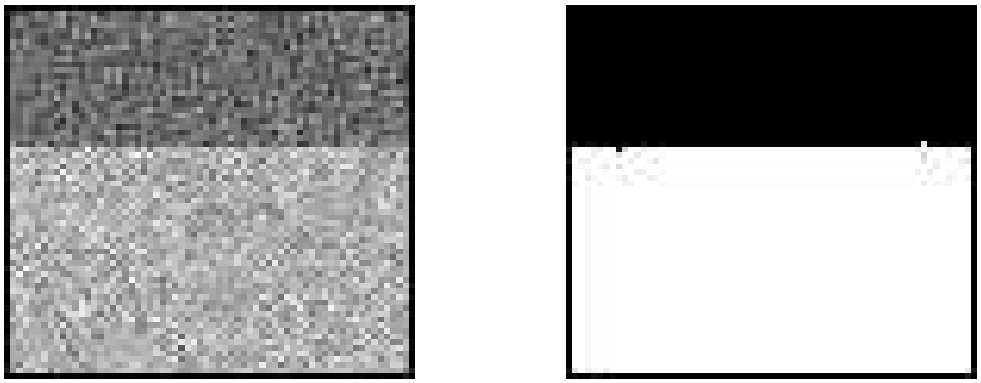

Figure 4: MPM segmentation of the image on the left, with the segmentation on the right.

– If decide to decrease, check to see if there is an empty class. If not, then reject. If yes, decide accept with probability the inverse of Equation 9. If accept, randomly select one of the empty classes and delete it from consideration.

As when R was fixed, there are two ways of applying this algorithm so that a segmentation can be found. The MAP solution would require this algorithm to be run in tandem with simulated annealing, in order to converge to a solution. That is to say, at the nth iteration of the algorithm, we use distributions proportional to $P(Y_s \mid X, R, \beta, \theta_1, \ldots, \theta_R, Y_{-s})^{1/T_n}$, $P(\beta \mid R, Y)^{1/T_n}$ and $P(\theta_r \mid X, Y)^{1/T_n}$ for a sequence of temperatures T_n converging to 0. The other possibility is the MPM, but it is not clear that that makes sense here, since the number of classes is changing and so the properties of the class assigned to a particular label can change. This is a common problem in mixture modelling. The MPM solution would make sense if the algorithm converges to a fixed number of classes, and we look at the labelling from that point forward.

Figures 4 and 5 show the analysis of a simple image composed of two Gaussian MRF textures, starting the algorithm with 3 classes. Two hundred iterations were computed. The algorithm was initialised by choosing pixel classes at random, letting $\beta = 0$, fixing the Gaussian MRF means to be 128.0, the clique parameters to 0 (which are prior mean values) and the variances to be 100.0^2, which represents a texture with large variance. Our experience is that the solution does not depend on these initial values. The algorithm successfully determines the segmentation into two classes after about 30 iterations, and because of this we present the MPM solution, based on the last 100 iterations. Figure 5 shows the number of classes at each iteration, and shows that once the algorithm has settled on two classes, proposals to increase to 3 classes are sometimes accepted but the new class is quickly eliminated again. The figure also shows the number of pixels in each class at each iteration, and shows that the third class becomes empty quite quickly and remains empty.

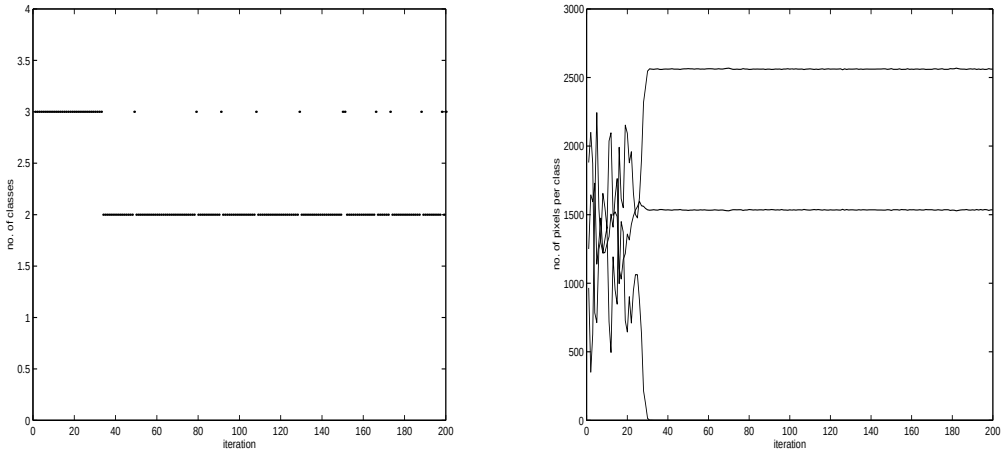

Figure 5: The number of classes at each iteration and the number of pixels in each class during the segmentation of the image in Figure 4.

Although the method has worked well in this very simple example, experiments with more complicated images shows that the method fails because there is not enough opportunity for eliminating classes and because the acceptance probability for new classes is not large. The problems seem to get worse as the number of classes in the image increases. Figure 6 is an example of an image containing 5 different MRF textures, segmented with this algorithm when a maximum of 8 textures are allowed. Over 5000 iterations, an empty class is never created and so the number of classes is never able to change. In fact, the algorithm is working exactly as the case where R was assumed to be 8.

From these and other experiments, we conclude that this approach can only be recommended when the number of classes is small, say 3 or 4. For a more generally useful algorithm, proposals that create or eliminate non-empty classes must be considered.

4.4 Allowing the Splitting and Merging of Non-Empty Classes

As soon as we consider changes in R that involve non-empty classes, the reversible jump scheme becomes a lot more complicated. We develop moves where R either increases or decreases by 1. Moves of any other type would be complicated to implement and obtaining reasonable acceptance probabilities would be difficult.

These moves are implemented by either splitting an existing class into two new classes, or merging two existing classes into one new class. The set of allowable moves of even this restricted class is very large. In [1], one such move was proposed. In the spirit of this paper, we will try to propose a somewhat simpler move, and investigate how well it performs.

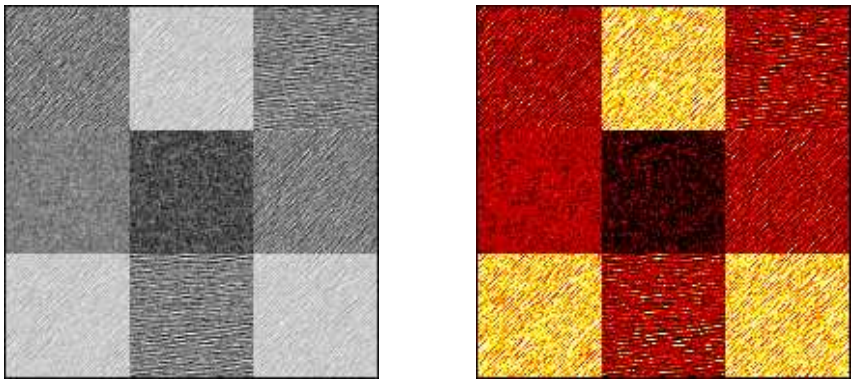

Figure 6: MPM segmentation of the image on the left (containing 5 GMRF textures), with the segmentation on the right (assuming a maximum of 8 classes), using the simple reversible jump scheme of creating and deleting empty classes.

The complete MCMC procedure consists of sampling from Y, Θ_R and β as in the fixed R case, then sampling R by either trying to increase or decrease R by 1. The increase is chosen with probability s_R, where

$$s_R = \begin{cases} 1, & \text{if } R = 1, \\ 0.5, & \text{if } 2 \leq R \leq R_{\max} - 1, \\ 0, & \text{if } R = R_{\max}. \end{cases}$$

and the decrease is chosen with probability $m_R = 1 - s_R$. These values are chosen to restrict R to be between 1 and $R_{\max}$, and to make about the same number of proposals to increase R as to decrease within this range.

The increase in R is accomplished by splitting an existing class into two new classes, and the decrease is accomplished by merging two existing classes. This move is then accepted or rejected. The whole procedure is then repeated for M iterations, where M is determined by the user.

To manage these moves better, we now order the labels by their mean value i.e. $\mu_r \leq \mu_{r+1}$. There are 3 reasons for the ordering. First, it will allow the mean of classes in the split and merge moves to be close, which increases the chance of proposing a reasonable move. Second, the ordering removes the non-identifiability of a segmentation, in the sense that, without the ordering, the method cannot distinguish between segmentations where the class labels are permuted. The non-identifiability would complicate the acceptance probabilities for the moves. Finally, the ordering allows us to easily produce segmentation plots where class number is identified with the mean intensity of the class. We will define the split and merge moves to preserve the ordering in means.

4.4.1 The Split Move

Following the general ideas presented above, we propose the following split move:

1. Randomly pick a non-empty class c to split.
2. Generate new parameters θ^*_{c1} and θ^*_{c2} from θ_c and random numbers u by a 1-1 transformation such that $\dim(\theta_c, u) = \dim(\theta^*_{c1}, \theta^*_{c2})$. In the case of GMRF texture models, we use the following, designed to be simple peturbations of the current parameters:

$$\begin{aligned} \mu^*_{c1} &= \mu_c - u_1\sigma_c & \mu^*_{c2} &= \mu_c + u_1\sigma_c \\ \sigma^{*2}_{c1} &= \sigma^2_c(1+u_2) & \sigma^{*2}_{c2} &= \sigma^2_c(1-u_2) \\ \theta^*_{c1,k} &= \theta_{ck} + u_{2+k} & \theta^*_{c2,k} &= \theta_{ck} - u_{2+k}, \end{aligned} \tag{10}$$

   where the u_1 is uniform(0,1), u_2 is uniform(-1,1), u_{2+k} is uniform(-0.1,0.1) and k is over the number of different clique parameters in the neighbourhood system of the GMRF.
3. Check to see that the new means are still ordered (that is, $\mu_{c-1} < \mu^*_{c1} < \mu^*_{c2} < \mu_{c+1}$) and lie in the prior range $[0, 255]$. If not, reject move immediately. If so, continue.
4. Assign all labels currently assigned to class c to classes $c1$ or $c2$. This is done as follows. Labels in the set $\mathcal{S}_c$ are assigned randomly to either $c1$ or $c2$. Then several Gibbs sampling sweeps are taken over $\mathcal{S}_c$ only, using the parameter values for $c1$ and $c2$ and the full conditionals of Equation 7. The Gibbs sampling hopefully produces homogeneous regions in $\mathcal{S}_c$ of each new class.
5. Compute the acceptance probability (see below) and accept or reject the move. If accept, relabel parameters and classes to include $c1$ and $c2$.

The choices for the u at step 2 allow the means of the new classes to be at most one current standard deviation from that of the old class, the variances to move between 0 and twice the current standard deviation, and does not allow large movements in clique parameter values (at most they change by 0.1). These moves are proposed after many tests of the method, and are a balance between allowing the possibility of reasonably large changes in the properties of old and new classes, and restricting changes so that the chance of accepting the move is not always 0.

The acceptance probability for a move that splits a class is still given by Equation 8. Each component in the probability is described below. We denote the current segmentation and parameters by Y and Θ_R, and proposed segmentation and parameters by Y^* and Θ^*_{R+1}:

1. We use the pseudo-likelihood approximation for the ratio $P(X, Y^* \mid \Theta^*_{R+1}, \beta)/P(X, Y \mid \Theta_R, \beta)$, thus:

$$\frac{P(X, Y^* \mid \Theta^*_{R+1}, \beta)}{P(X, Y \mid \theta_c, \beta)} \approx \frac{\prod_{s\in\mathcal{S}} P(x_s \mid x_j, j \neq s, \theta^*_{y^*_s})\, P(y^*_s \mid y^*_j, j \neq s, \beta)}{\prod_{s\in\mathcal{S}} P(x_s \mid x_j, j \neq s, \theta_c) \prod_{s\in\mathcal{S}} P(y_s \mid y_j, j \neq s, \beta)}$$
$$= \prod_{s\in\mathcal{S}_c} \frac{P(x_s \mid x_j, j \neq s, \theta^*_{y^*_s})\, P(y^*_s \mid y^*_j, j \neq s, \beta)}{P(x_s \mid x_j, j \neq s, \theta_c)\, P(y_s \mid y_j, j \neq s, \beta)} \quad (11)$$

these are just the full conditional densities for each pixel colour and label. However we note that the term $P(y^*_s \mid y^*_j, j \neq s, \beta)$ will be:

$$\frac{\exp\left(\beta \sum_{j\in\eta_s} \delta(y^*_s - y^*_j)\right)}{\sum_{r=1}^{R+1} \exp\left(\beta \sum_{j\in\eta_s} \delta(r - y^*_j)\right)},$$

whereas

$$P(y_s \mid y_j, j \neq s, \beta) = \frac{\exp\left(\beta \sum_{j\in\eta_s} \delta(y_s - y_j)\right)}{\sum_{r=1}^{R} \exp\left(\beta \sum_{j\in\eta_s} \delta(r - y_j)\right)},$$

so they have normalising constants over the new and old classes respectively.

2. The prior on the mean is uniform(0,255), on the clique parameters is uniform over the allowable range, and the prior on the variances is inverse gamma (see Section 2.2 for the justification of this). Therefore the prior ratio is:

$$\frac{P(\Theta^*_{R+1})}{P(\Theta_R)} = \frac{\prod_{r=1}^{R+1} P(\theta^*_r)}{\prod_{r=1}^{R} P(\theta_r)} = \frac{P(\theta^*_{c1})\, P(\theta^*_{c2})}{P(\theta_c)}$$
$$= \frac{1}{255} \times \frac{a^b}{\Gamma(b)} \left(\frac{\sigma_c^2}{\sigma^{*2}_{c1}\sigma^{*2}_{c2}}\right)^{b+1} \exp\left(a\left(\frac{1}{\sigma_c^2} - \frac{1}{\sigma^{*2}_{c1}} - \frac{1}{\sigma^{*2}_{c2}}\right)\right) \times |A_\theta|, \quad (12)$$

where a and b are the prior hyperparameters for the variance and $|A_\theta|$ is the volume of the allowable range of the clique parameters. The conditions on the clique parameters are given in [7] for the second order case, from which $|A_\theta|$ can be estimated.

3. The ratio $P(R+1)/P(R)$ would be $(1-p)$ if R is geometric with parameter p, or $\lambda/(R+1)$ if R is Poisson with mean λ, for $R < R_{\max}$, and 0 otherwise.

4. The density $p(u)$ is the product of a uniform(0,1), a uniform(-1,1) and several uniform (-0.1,0.1) densities (one for each clique parameter), thus $p(u) = 1 \times 0.5 \times \prod_k 5$.

5. $P(\text{decrease } R+1) = m_{R+1}/(1 + \text{no. of non-empty classes})$, since it is the probability m_{R+1} of choosing to decrease the number of classes (1 if $R = R_{\max}$, 0.5 if $2 \leq R < R_{\max}$ and 0 if $R = 1$) and picking one class (which we merge with a neighbouring class with the closest mean).

6. $P(\text{increase } R)$ is (s_R/no. of non-empty classes) $\times P_{\text{alloc}}$, since it is the probability s_R of choosing to split a class (1 if $R = 1$, 0.5 if $2 \leq R < R_{\max}$ and 0 if $R = R_{\max}$) times the random choice of a non-empty class to split, then the class labels are reassigned to $c1$ and $c2$ with a probability P_{alloc}. This latter is simply $P(y^* \mid x, \Theta^*_{R+1}\beta)$ restricted to $\mathcal{S}_c$ and given that y^*_s can only be $c1$ or $c2$. We approximate by the pseudolikehood:

$$P_{\text{alloc}} = \prod_{s \in \mathcal{S}_c} \frac{P(y^*_s \mid x, y^*_j, j \neq s, \theta^*_{y^*_s}, \beta)}{P(y^*_s = c1 \mid x, y^*_j, j \neq s, \theta^*_{c1}, \beta) + P(y^*_s = c2 \mid x, y^*_j, j \neq s, \theta^*_{c2}, \beta)}; \quad (13)$$

each term in the product is given by Equation 7.

7. The Jacobian of the transformation in Equation 10 is $4\sigma_c^3 2^K$ (recall K is the number of clique parameters).

The above list of terms are multiplied together to give the acceptance probability. For example, if we choose a geometric prior for R with parameter p and second order GMRFs for the textures, we have $P(\text{accept}) = \min\{1, A\}$, where

$$\begin{aligned} A = \Bigg\{ \prod_{s \in \mathcal{S}_c} & \frac{\exp\left[-\frac{1}{2\sigma^{*2}_{y^*_s}}\left\{x_s - \mu^*_{y^*_s} - \sum_{k=1}^4 \sum_{j|<s,j>_k} \theta^*_{y^*_s,k}(x_j - \mu^*_{y^*_s})\right\}^2\right] \sqrt{2\pi\sigma_c^2}}{\exp\left[-\frac{1}{2\sigma_c^2}\left\{x_s - \mu_c - \sum_{k=1}^4 \sum_{j|<s,j>_k} \theta_{ck}(x_j - \mu_c)\right\}^2\right] \sqrt{2\pi\sigma^{*2}_{y^*_s}}} \\ & \times \frac{\exp\left[\beta \sum_{j \in \eta_s} \delta(y^*_s - y^*_j)\right] \sum_{r=1}^R \exp\left[\beta \sum_{j \in \eta_s} \delta(r - y_j)\right]}{\exp\left[\beta \sum_{j \in \eta_s} \delta(y_s - y_j)\right] \sum_{r=1}^{R+1} \exp\left[\beta \sum_{j \in \eta_s} \delta(r - y^*_j)\right]} \Bigg\} \\ & \times \frac{|A_\theta|}{255} \frac{a^b}{\Gamma(b)} \left(\frac{\sigma_c^2}{\sigma^{*2}_{c1}\sigma^{*2}_{c2}}\right)^{b+1} \exp\left(a\left(\frac{1}{\sigma_c^2} - \frac{1}{\sigma^2_{c1}} - \frac{1}{\sigma^2_{c2}}\right)\right) \times (1-p) \\ & \times \frac{1}{1 \times 0.5 \times 5^4} \times \frac{m_{R+1} \times \text{no. of non-empty classes}}{s_R \times (1 + \text{no. of non-empty classes})\, P_{\text{alloc}}} \times 64\sigma_c^3, \quad (14) \end{aligned}$$

where

$$P_{\text{alloc}} = \prod_{s \in \mathcal{S}_c} \frac{\exp\left[-\frac{1}{2\sigma^{*2}_{y^*_s}}\left\{x_s - \mu^*_{y^*_s} - \sum_{k=1}^K \sum_{j|<s,j>_k} \theta^*_{y^*_s,k}(x_j - \mu^*_{y^*_s})\right\}^2 + \beta \sum_{j \in \eta_s} \delta(y^*_s - y^*_j)\right]}{C_s \times \sqrt{2\pi\sigma^{*2}_{y^*_s}}}, \quad (15)$$

for normalising constant summing over the two new classes:

$$C_s = \sum_{r=c1,c2} \frac{\exp\left[-\frac{1}{2\sigma^{*2}_r}\left\{x_s - \mu^*_r - \sum_{k=1}^K \sum_{j|<s,j>_k} \theta^*_{rk}(x_j - \mu^*_r)\right\}^2 + \beta \sum_{j \in \eta_s} \delta(r - y^*_j)\right]}{\sqrt{2\pi\sigma^{*2}_r}}.$$

Note that several terms in A cancel out, somewhat simplifying the expression, but we have left it in this form so as it is clear where each term in the expression has come from.

4.4.2 The Merge Move

A merge move is to choose a class $c1$, select the class $c2$ that has closest mean to it (from the ordering on means property, this is either $c1+1$ or $c1-1$) and propose new parameters for the merged class c that are the inverse of the transformation in Equation 10. We use the following transformation of parameters for the reversal of the split transformation:

$$\mu_c^* = \frac{n_1}{n_1+n_2}\mu_{c1} + \frac{n_2}{n_1+n_2}\mu_{c2}; \quad \sigma_c^{*2} = \frac{\sigma_{c1}^2 + \sigma_{c2}^*}{2}; \quad \theta_{ck}^* = \frac{\theta_{c1,k} + \theta_{c2,k}}{2}, \tag{16}$$

where n_1 and n_2 are the number of pixels assigned to $c1$ and $c2$ respectively. The Jacobian of this transformation is

$$\frac{1}{2\sqrt{2}\sqrt{\sigma_{c1}^2 + \sigma_{c2}^2}} \frac{1}{2^K}.$$

If the old segmentation and parameters are Y and Θ_R, and the new ones are Y^* and Θ_{R-1}^*, then the accept probability when we have a geometric prior on R with parameter p and 2nd order GMRF textures is the minimum of 1 and A, where

$$\begin{aligned} A = \Bigg\{ \prod_{s\in\mathcal{S}_c} & \frac{\exp\left[-\frac{1}{2\sigma_c^{*2}}\left\{x_s - \mu_c^* - \sum_{k=1}^4 \sum_{j|<s,j>_k} \theta_{ck}^*(x_j - \mu_c^*)\right\}^2\right] \sqrt{2\pi\sigma_{y_s}^2}}{\exp\left[-\frac{1}{2\sigma_{y_s}^2}\left\{x_s - \mu_{y_s} - \sum_{k=1}^4 \sum_{j|<s,j>_k} \theta_{y_s,k}(x_j - \mu_{y_s})\right\}^2\right] \sqrt{2\pi\sigma_c^{*2}}} \\ & \times \frac{\exp\left[\beta \sum_{j\in\eta_s} \delta(y_s^* - y_j^*)\right] \sum_{r=1}^R \exp\left[\beta \sum_{j\in\eta_s} \delta(r - y_j)\right]}{\exp\left[\beta \sum_{j\in\eta_s} \delta(y_s - y_j)\right] \sum_{r=1}^{R-1} \exp\left[\beta \sum_{j\in\eta_s} \delta(r - y_j^*)\right]} \Bigg\} \\ & \times \frac{255}{|A_\theta|} \frac{\Gamma(b)}{a^b} \left(\frac{\sigma_{c1}^2\sigma_{c2}^2}{\sigma_c^{*2}}\right)^{b+1} \exp\left(a\left(\frac{1}{\sigma_{c1}^2} + \frac{1}{\sigma_{c2}^2} - \frac{1}{\sigma_c^{*2}}\right)\right) \times \frac{1}{(1-p)} \\ & \times 0.5 \times 5^4 \times \frac{s_{R-1} \times \text{no. of non-empty classes} \times P_{\text{alloc}}}{m_R \times (\text{no. of non-empty classes} - 1)} \times \frac{1}{32\sqrt{2}\sqrt{\sigma_{c1}^2 + \sigma_{c2}^2}}, \end{aligned} \tag{17}$$

where P_{alloc} is given by Equation 15, thinking of the reverse split move from y^* to y. If the move is accepted, the classes $c1$ and $c2$ are eliminated and replaced by c.

4.4.3 Combining the Moves with Simulated Annealing

If we wish to find the MAP labelling, the MCMC algorithm is at the nth iteration to sample from a distribution proportional to $P(Y, R, \Theta_R, \beta \mid X)^{1/T_n}$, where T_n is a decreasing sequence

of temperatures that converges to 0. In this case, each element of the likelihood and prior ratio terms in the acceptance probability are raised to $1/T_n$. If we use the distribution raised to $1/T_n$ for the reallocation of classes in a split move, P_{alloc} is also changed in this way. The Jacobian term, m_R, s_R and $P(u)$ remain unchanged.

4.4.4 Experiments

First we attempt to segment the simple image of Figure 4, starting with 5 classes and allowing a maximum of 10 classes. Figure 7 is a summary of application of the above algorithm to find the MAP solution. A geometric prior for R with a mean of 5 was assumed. We see that the algorithm does well, converging down to 2 classes and never allowing more than 6. It gives exactly the same solution as the simple RJ approach of Section 4.2. The algorithm was run for 1500 iterations, and the figure also shows how the number of non-empty classes and the number of pixels per class varied as a function of iterations.

Next we look at an aerial image. Figure 8 shows an aerial image of an agricultural area, with the MAP solution from the reversible jump algorithm. The algorithm finally converged to 10 classes. This is an oversegmentation, in the sense of discriminating the main classes in the image: different fields, trees and the road. It has also segmented each field into different classes that correspond to different colourations in the image.

We conclude with an analysis of the satellite image of an area of Holland whose analysis for $R = 4$ is given in Figure 2. In this case we end up with 24 classes. This represents another oversegmentation. The image consists of fields, each composed of very few grey levels, and the algorithm classifies the fields of different grey levels into different classes, and assigns 3 classes to the more textured urban areas. These three urban classes have different mean intensity, so the algorithm seems to divide the urban areas according to intensity, and has ignored the small and similar scale texture of all the urban areas. Although there are a large number of classes, most of these are assigned to a small number of pixels. In fact, 90% of pixels are assigned to only 4 classes. In spite of this, the algorithm did not merge the classes assigned to only a small number of pixels.

The final comment is on the speed of this approach. Clearly the split/merge sampler is slower than the algorithm where R is known. The most computationally intensive move is the split, and the time taken for it depends on how large is the class to be split. As a general rule, in these examples, where the number of classes ranged between 10 and 30, the split/merge sampler took about 50% longer than the sampler where R was known and the number of classes was roughly the same.

5 Conclusions

In this paper we have developed a series of segmentation algorithms based around Markov random fields and Bayesian inference, implemented with MCMC. The simplest approach is to assume a known number of classes, and this method is shown to work when this number can be specified well. The more general case of an unknown number of classes is more

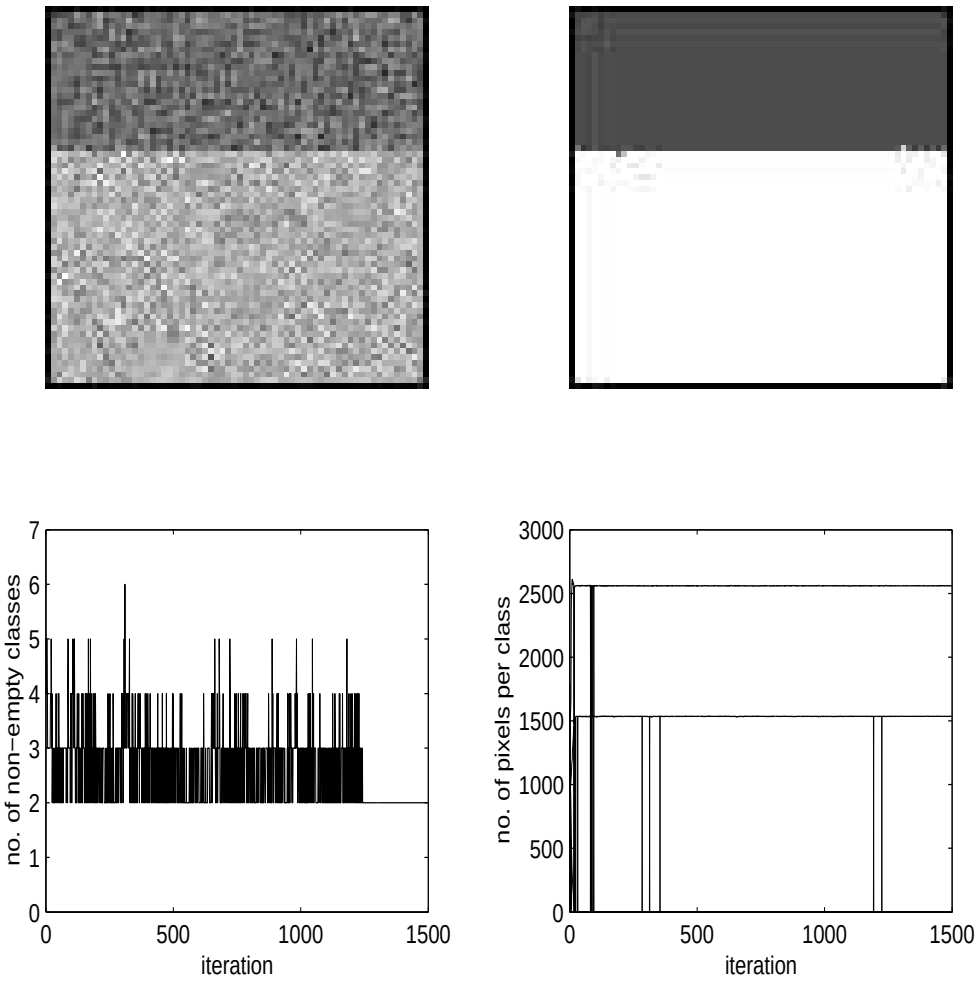

Figure 7: MAP segmentation of the image in the upper left, using splitting and merging of non-empty classes, also showing the number of non-empty classes and number of pixels per class as a function of iterations of the algorithm.

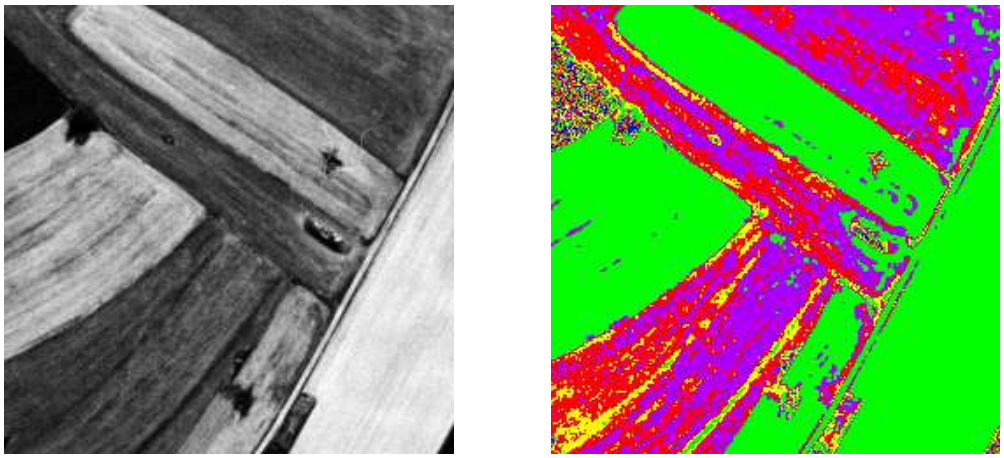

Figure 8: An aerial image on the left, with MAP segmentation, using splitting and merging of non-empty classes.

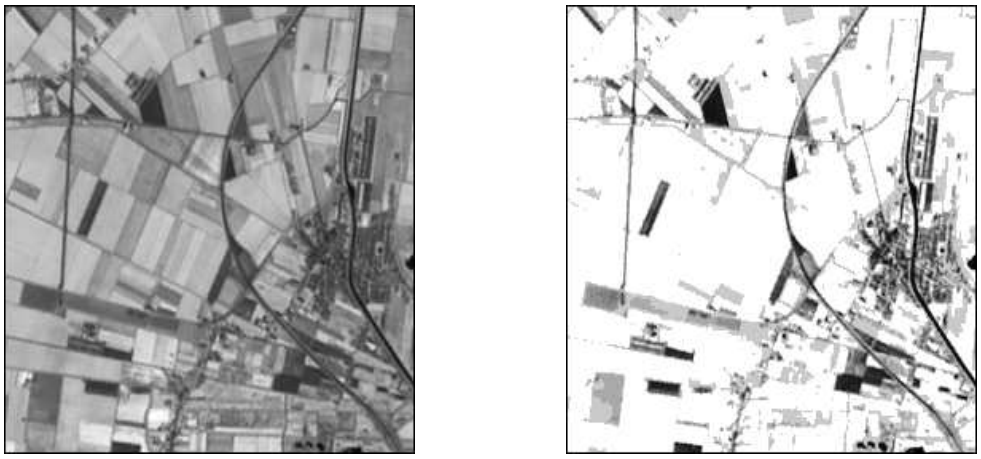

Figure 9: A satellite image on the left, with MAP segmentation, using splitting and merging of non-empty classes.

difficult to treat in the Bayesian framework, and we must resort to reversible jump MCMC sampling. Two such samplers were investigated. From the examples given here and others that have been done, we conclude that the simple birth/death reversible jump sampler is only applicable to images with very few classes, since the algorithm does not produce empty classes often enough for them to be eliminated. The more complex split/merge moves seem to oversegment satellite images. Typically there are several classes that are assigned to only a small proportion of the image, and yet are not merged.

Analysis of the acceptance probabilities for the split and merge moves shows that they are either 1 or very small, and are completely dominated by the relative sizes of the likelihood ratio and P_{alloc}. Thus a stronger prior on R would not alter the number of classes found. Other possibilities for reducing the number of classes found would be larger neighbourhood orders for the labels, or fixing β to be reasonably large and running an MPM sampler. Other approaches that infer the number of classes may also provide ideas: those using stochastic geometry and reversible jump MCMC (see [5] for work in Projet Ariana) and those based on mean field EM and the BIC criterium (see [11]). These will be topics for future work.

Acknowledgements

This work was done while the first author was a visiting researcher at Projet Ariana in INRIA Sophia Antipolis during April – July 2000, supported by the European Science Foundation "Highly Structured Stochastic Systems" Programme, and has subsequently formed part of the European Union Research and Training Network "MOUMIR". The authors would like to thank the French Space Agency (CNES) for providing SPOT satellite images and the French Mapping Institute (IGN) for providing the aerial data.

References

[1] S. A. Barker and P. J. W. Rayner. Unsupervised image segmentation using Markov random field models. *Pattern Recognition*, 33:587–602, 200.

[2] J. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Roy. Statist. Soc. B*, 36:192–236, 1974.

[3] S. Geman and D. Geman. Stochastic relaxation, Gibb's distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741, 1984.

[4] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[5] M. Imberty and X. Descombes. Simulation de processus objets: etude de faisabilité pour une application à la segmentation d'image. INRIA Research Report 3881, 2000.

[6] Z. Kato. Bayesian color image segmentation using reversible jump Markov chain Monte Carlo. Technical Report 01/99-R055, European Research Consortium for Informatics and Mathematics, 1999.

[7] S. Lakshmanan and H. Derin. Valid parameter space for 2-D Gaussian Markov random fields. *IEEE Trans. Inform. Theory*, 39:703–709, 1993.

[8] D. E. Melas. *A Bayesian Approach to the Segmentation of Textured Images.* PhD thesis, Statistics Department, University of Dublin, 1999.

[9] D. E. Melas and S. P. Wilson. Texture based image segmentation using the double MRF model. In K. V. Mardia, C. A. Gill, and R. G. Aykroyd, editors, *Proceedings in the Art and Science of Bayesian Image Analysis*, pages 169–175, Leeds, 1997. Leeds University Press.

[10] D. E. Melas and S. P. Wilson. Double Markov random fields and Bayesian image segmentation. *To appear, IEEE Trans. Sig. Proc.*, 2002.

[11] N. Peyrard. Approximation de type champ moyen des modèles de champ de Markov pour la segmentation de données spatiales. *PhD Thesis, Joseph Fourier University, Grenoble, France*, 2001.

[12] C. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer, New York, 1999.

[13] J. Zhang, J. W. Modestino, and D. A. Langan. Maximum likelihood parameter estimation for unsupervised stochastic model-based image segmentation. *IEEE Trans. Image Processing*, 3:404–419, 1994.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399