Reversible Jump Markov Chain Monte Carlo for Unsupervised MRF Color Image Segmentation *

Zoltan Kato

University of Szeged, Institute of Informatics, P.O. Box 652, H-6701 Szeged, Hungary,

Fax:+36 62 546 397, Email: kato@inf.u-szeged.hu

Abstract

Reversible jump Markov chain Monte Carlo (RJMCMC) is a recent method which makes it possible to construct reversible Markov chain samplers that jump between parameter subspaces of different dimensionality. In this paper, we propose a new RJMCMC sampler for multivariate Gaussian mixture identification and we apply it to color image segmentation. For this purpose, we consider a first order Markov random field (MRF) model where the singleton energies derive from a multivariate Gaussian distribution and second order potentials favor similar classes in neighboring pixels. The proposed algorithm finds the most likely number of classes, their associated model parameters and generates a segmentation of the image by classifying the pixels into these classes. The estimation is done according to the Maximum A Posteriori (MAP) criterion. Experimental results are promising, we have obtained accurate results on a variety of real color images.

1 Introduction

MRF modeling and MCMC methods are successfully used in different areas of image processing. In fact, the simplest statistical model for an image consists of the probabilities of pixel classes. The knowledge of the dependencies between nearby pixels can be modeled by a MRF. Such models are much more powerful [7, 12], even if it is not easy to determine the values of the parameters which specify a MRF. If each pixel class is represented by a different model then the observed image may be viewed as a sample from a realization of an underlying label field. Unsupervised segmentation can therefore be treated as an *incomplete data problem* where the color values are observed, the label field is missing and the associated class model parameters, *including the number of classes*, need to be estimated. Such problems are often solved using MCMC procedures. Although the general theory and methodology of these algorithms are fairly standard, they have their limitations in case of problems with parameters of varying dimension. Recently, a novel method, called Reversible Jump MCMC (RJMCMC), has been proposed by Green [5].

*This research was partially supported by an ERCIM postdoctoral fellowship during the author's stay at CWI, Amsterdam; the Janos Bolyai research fellowship of the Hungarian Academy of Sciences; and the Hungarian Scientific Research Fund – OTKA T046805.

This method makes it possible to construct reversible Markov chain samplers that jump between parameter subspaces of different dimensionality. In this paper, we will develop a RJMCMC sampler for identifying multi-variate Gaussian mixtures. In particular, we will apply this technique to solve the unsupervised color image segmentation problem in a Markovian framework.

Due to the difficulty of estimating the number of pixel classes (or clusters), unsupervised algorithms often assume that this parameter is *known a priori*. When the number of pixel classes is also being estimated, the unsupervised segmentation problem may be treated as a model selection problem over a combined model space. Basically, there are two approaches in the literature. One of them is an exhaustive search of the combined parameter space[15, 9]: Segmentation and parameter estimates are obtained via an iterative algorithm by alternately sampling the label field based on the current estimates of the parameters. Then the maximum likelihood estimates of the parameter values are computed using the current labeling. The resulting estimates are then applied to a model fitting criterion to select the optimum number of classes. Another approach consists of a two step approximation technique [7, 10]: the first step is a coarse segmentation of the image into the most likely number of regions. Then the parameter values are estimated from the resulting segmentation and the final result is obtained via a supervised segmentation.

Our approach consists of building a Bayesian color image model using a first order MRF. The observed image is represented by a mixture of multivariate Gaussian distributions while inter-pixel interaction favors similar labels at neighboring sites. In a Bayesian framework [4], we are interested in the *posterior distribution* of the unknowns given the observed image. Herein, the unknowns comprise the hidden label field configuration, the Gaussian mixture parameters, the MRF hyperparameter, and the number of mixture components (or classes). Then a RJMCMC algorithm is used to sample from the whole posterior distribution in order to obtain a MAP estimate via simulated annealing [4]. Until now, RJMCMC has been applied to univariate Gaussian mixture identification [13] and its applications in different areas like inference in hidden Markov models [14], intensity based image segmentation [1], and computing medial axes of 2D shapes [16]. The novelty of our approach is two-fold: first, we extend the ideas in [13, 1] and we propose a RJMCMC method for identifying multi-variate Gaussian mixtures. Second, we apply it to unsupervised color image segmentation: RJMCMC allows us the direct sampling of the whole posterior distribution defined over the combined model space thus reducing the optimization process to a single simulated annealing run. Another advantage is that no coarse segmentation neither exhaustive search over a parameter subspace is required. Although for clarity of presentation we will concentrate on the case of three-variate Gaussians in the remaining part of this paper, it is straightforward to extend the equations to higher dimensions.

2 Color Image Segmentation Model

Let us suppose that the observed image $\mathscr{F} = \{\vec{f}_s | s \in \mathscr{S}, \forall i : 0 < \vec{f}_s^{\,i} < 1\}$ consists of three spectral component values at each pixel s denoted by the vector $\vec{f}_s$. The segmentation is done by assigning a label ω_s from the set of labels $\Lambda = \{1, 2, \ldots, L\}$ to each site s. $\omega \in \Omega$ denotes a labeling (or segmentation), Ω is the set of all possible labeling.

Basically, we regard our image as a sample drawn from a first order MRF. The goal of our analysis is inference about the number L of Gaussian mixture components, the component parameters $\Theta = \{\Theta_\lambda = (\vec{\mu}_\lambda, \Sigma_\lambda) \| \lambda \in \Lambda\}$, the component weights p_λ, summing to 1, the clique potential (or inter-pixel interaction strength) β, and the segmentation ω. The joint distribution of the variables $L, p, \beta, \omega, \Theta, \mathscr{F}$ is given by:

$$P(L,p,\beta,\omega,\Theta,\mathscr{F}) = P(\omega,\mathscr{F} \mid \Theta,\beta,p,L) P(\Theta,\beta,p,L) \tag{1}$$

In our context, it is natural to impose conditional independences on (Θ,β,p,L) so that their joint probability reduces to the product of priors:

$$P(\Theta,\beta,p,L) = P(\Theta)P(\beta)P(p)P(L) \tag{2}$$

Let us concentrate now on the posterior distribution of $(\mathscr{F},\omega)$ which may be expressed as: $P(\mathscr{F} \mid \omega,\Theta,\beta,p,L)P(\omega \mid \Theta,\beta,p,L)$. Before further proceeding, let us examine the above factorization. As we declared earlier, pixel classes are represented by a three-variate Gaussian distribution and the underlying MRF label process follows a Gibbs distribution defined over a first order neighborhood system [4]. Thus, we can impose further conditional independences yielding:

$$\begin{aligned}
P(\mathscr{F} \mid \omega,\Theta,\beta,p,L) &= P(\mathscr{F} \mid \omega,\Theta) && (3)\\
&= \prod_{s\in\mathscr{S}} \frac{1}{\sqrt{(2\pi)^3 \mid \Sigma_{\omega_s} \mid}} \exp\left(-\frac{1}{2}(\vec{f}_s - \vec{\mu}_{\omega_s})\Sigma_{\omega_s}^{-1}(\vec{f}_s - \vec{\mu}_{\omega_s})^T\right) \\
P(\omega \mid \Theta,\beta,p,L) &= P(\omega \mid \beta,p,L) && (4)\\
&= \frac{1}{Z(\beta,p,L)} \exp(-U(\omega \mid \beta,p,L)) \text{ , where} \\
U(\omega \mid \beta,p,L) &= \sum_{s\in\mathscr{S}} -\log(p_{\omega_s}) + \beta \sum_{\{s,r\}\in\mathscr{C}} \delta(\omega_s,\omega_r) && (5)
\end{aligned}$$

$U(\omega \mid \beta,p,L)$ is called the *energy function*. $\delta(\omega_s,\omega_r) = 1$ if ω_s and ω_r are different and -1 otherwise. $Z(\beta,p,L) = \sum_{\omega\in\Omega} \exp(-U(\omega \mid \beta,p,L))$ denotes the normalizing constant (or *partition function*). Furthermore, $\mathscr{C}$ denotes the set of cliques and $\{s,r\}$ is a doubleton containing the neighboring pixel sites s and r. Since the partition function $Z(\beta,p,L)$ is not tractable [8], the comparison of the likelihood of two differing MRF realizations from Eq. (5) is infeasible. However, we can compare their Pseudo-Likelihood [8]. Therefore, using Eq. (1), Eq. (2), Eq. (4), and the fact that $P(\mathscr{F})$ is constant for a given image, we can now easily approximate the posterior distribution:

$$\begin{aligned}
P(L,p,\beta,\omega,\Theta \mid \mathscr{F}) \;\approx\; & P(\mathscr{F} \mid \omega,\Theta) \prod_{s\in\mathscr{S}} \frac{p_{\omega_s} \exp\left(-\beta \sum_{\forall r:\{s,r\}\in\mathscr{C}} \delta(\omega_s,\omega_r)\right)}{\sum_{\lambda\in\Lambda} p_\lambda \exp\left(-\beta \sum_{\forall r:\{s,r\}\in\mathscr{C}} \delta(\lambda,\omega_r)\right)} \\
& \times P(\beta)P(L) \prod_{\lambda\in\Lambda} P(\vec{\mu}_\lambda)P(\Sigma_\lambda)P(p_\lambda) \qquad (6)
\end{aligned}$$

In order to simplify our presentation, we will follow [13] and chose uniform reference priors for $L, \vec{\mu}_\lambda,\ \Sigma_\lambda, p_\lambda$ $(\lambda \in \Lambda)$. However, we note that informative priors could improve the quality of estimates, especially in the case of the number of classes.

3 Sampling from the Posterior Distribution

A broadly used tool to sample from the posterior distribution in Eq. (6) is the Metropolis-Hastings method [6]. Classical methods, however, can not be used due to the changing dimensionality of the parameter space. To overcome this limitation, a promising approach, called Reversible Jump MCMC (RJMCMC), has been proposed in [5]. When we have multiple parameter subspaces of different dimensionality, it is necessary to devise different *move types* between the subspaces [5]. These will be combined in a so called *hybrid sampler*. For our image segmentation model, we shall make use of the following move types:

1. sampling the labels ω (ie. re-segment the image);
2. sampling Gaussian parameters $\Theta = \{(\vec{\mu}_\lambda, \Sigma_\lambda)\}$;
3. sampling the mixture weights $p_\lambda (\lambda \in \Lambda)$;
4. sampling the MRF hyperparameter β;
5. sampling the number of classes L (splitting one mixture component into two, or combining two into one).

We note that the only randomness in scanning these move types is the random choice between splitting and merging in move (5). One iteration of the hybrid sampler, also called a *sweep*, consists of a complete pass over these moves. The first four move types are conventional in the sense that they do not alter the dimension of the parameter space. In each of these move types, the posterior distribution can be easily derived from Eq. (6) by setting unaffected parameters to their current estimate. For example, in move (1), the parameters L, p, β, Θ are set to their estimates $\widehat{L}$, $\widehat{p}$, $\widehat{\beta}$, $\widehat{\Theta}$. Thus the posterior in Eq. (6) reduces to the following form:

$$\begin{aligned} P(L,p,\beta,\omega,\Theta \mid \mathscr{F}) &\approx P(\mathscr{F} \mid \omega, \widehat{\Theta}) P(\omega \mid \widehat{\beta}, \widehat{p}, \widehat{L}) \\ &\approx \prod_{s \in \mathscr{S}} \left(\frac{1}{\sqrt{(2\pi)^3 \mid \widehat{\Sigma}_{\omega_s} \mid}} \exp\left(-\frac{1}{2} (\vec{f}_s - \widehat{\vec{\mu}}_{\omega_s}) \widehat{\Sigma}_{\omega_s}^{-1} (\vec{f}_s - \widehat{\vec{\mu}}_{\omega_s})^T \right) \right) \\ &\quad \times \prod_{s \in \mathscr{S}} \widehat{p}_{\omega_s} \exp\left(-\widehat{\beta} \sum_{\forall r : \{s,r\} \in \mathscr{C}} \delta(\omega_s, \omega_r) \right) \end{aligned} \tag{7}$$

where $\widehat{L}$, $\widehat{p}$, $\widehat{\beta}$, $\widehat{\Theta}$ are the current estimates of the parameters L, p, β, Θ. Basically, the above equation corresponds to a segmentation with *known parameters*.

Hereafter, we will focus on move (5), which requires the use of the reversible jump mechanism. This move type involves changing L by 1 and making necessary corresponding changes to ω, Θ and p. We remark that at this move type, basically the whole posterior distribution in Eq. (6) is sampled. Only β can be set to its estimate $\widehat{\beta}$.

3.1 Splitting One Class into Two

The split proposal begins by choosing a class λ *at random* with a uniform probability $P_{select}^{split}(\lambda) = 1/L$. Then L is increased by 1 and λ is split into λ_1 and λ_2. In doing so, a new set of parameters need to be generated. Altering L changes the dimensionality of the variables Θ and p. Thus we shall define a deterministic function ψ as a function of these Gaussian mixture parameters:

$$(\Theta^+, p^+) = \psi(\Theta, p, u) \tag{8}$$

where the superscript $+$ denotes parameter vectors after increasing L. u is a set of random variables having as many elements as the degree of freedom of joint variation of the current parameters (Θ, p) and the proposal (Θ^+, p^+). Note that this definition satisfies the *dimension matching* constraint [5], which guarantees that one can jump back and forth between different parameter sub-spaces. The new parameters of λ_1 and λ_2 are assigned by matching the 0^{th}, 1^{th}, 2^{th} moments of the component being split to those of a combination of the two new components [13]:

$$p_\lambda = p^+_{\lambda_1} + p^+_{\lambda_2} \tag{9}$$

$$p_\lambda \vec{\mu}_\lambda = p^+_{\lambda_1} \vec{\mu}^+_{\lambda_1} + p^+_{\lambda_2} \vec{\mu}^+_{\lambda_2} \tag{10}$$

$$p_\lambda (\vec{\mu}_\lambda \vec{\mu}^T_\lambda + \Sigma_\lambda) = p^+_{\lambda_1} (\vec{\mu}^+_{\lambda_1} \vec{\mu}^{+T}_{\lambda_1} + \Sigma^+_{\lambda_1}) + p^+_{\lambda_2} (\vec{\mu}^+_{\lambda_2} \vec{\mu}^{+T}_{\lambda_2} + \Sigma^+_{\lambda_2}) \tag{11}$$

There are 10 degrees of freedom in splitting λ since covariance matrices are symmetric. Therefore, we need to generate a random variable $u1$, a random vector $\vec{u2}$ and a symmetric random matrix $u3$. We can now compute the new parameter values as follows:

$$p^+_{\lambda_1} = p_\lambda u1, \tag{12}$$

$$p^+_{\lambda_2} = p_\lambda (1 - u1) \tag{13}$$

$$\mu^+_{\lambda_1,i} = \mu_{\lambda,i} + u2_i \sqrt{\Sigma_{\lambda,i,i} \frac{1 - u1}{u1}} \tag{14}$$

$$\mu^+_{\lambda_2,i} = \mu_{\lambda,i} - u2_i \sqrt{\Sigma_{\lambda,i,i} \frac{u1}{1 - u1}} \tag{15}$$

$$\Sigma^+_{\lambda_1,i,j} = \begin{cases} u3_{i,i} \left(1 - u2_i^{\,2}\right) \Sigma_{\lambda,i,i} \dfrac{1}{u1} & \text{if } i = j \\ u3_{i,j} \Sigma_{\lambda,i,j} \sqrt{\left(1 - u2_i^{\,2}\right)} \sqrt{\left(1 - u2_j^{\,2}\right) u3_{i,i} u3_{j,j}} & \text{if } i \neq j \end{cases} \tag{16}$$

$$\Sigma^+_{\lambda_2,i,j} = \begin{cases} \left(1 - u3_{i,i}\right) \left(1 - u2_i^{\,2}\right) \Sigma_{\lambda,i,i} \dfrac{1}{u1} & \text{if } i = j \\ \left(1 - u3_{i,j}\right) \Sigma_{\lambda,i,j} & \\ \quad \times \sqrt{\left(1 - u2_i^{\,2}\right) \left(1 - u2_j^{\,2}\right)} \sqrt{\left(1 - u3_{i,i}\right) \left(1 - u3_{j,j}\right)} & \text{if } i \neq j \end{cases} \tag{17}$$

We note that in Eq. (14) and Eq. (15), we randomly alternate the $+$ and $-$ signs of $u2_i$ for each i. This is needed to get proper mean values after splitting. The random variables u are chosen from the interval $(0, 1]$. In order to favor splitting a class into roughly equal portions, beta$(1.1, 1.1)$ distributions are used. To guarantee numerical stability in inverting $\Sigma^+_{\lambda_1}$ and $\Sigma^+_{\lambda_2}$, one can use some regularization like in [3], or one can use the well-known Wishart distribution [11]. However, we did not experience such problems, mainly because the obtained covariance matrices are also reestimated from the image

Original image Segmentation result (3 classes)

Figure 1: Segmentation of image *rose41*.

data in subsequent move types. Therefore as long as our input image can be described by a mixture of Gaussians, we can expect that the estimated covariance matrices are correct.

The next step is the reallocation of those sites $s \in \mathscr{S}_\lambda$ where $\omega_s = \lambda$. This reallocation is based on the new parameters and has to be completed in such a way as to ensure the resulting labeling ω^+ is drawn from the posterior distribution with $\Theta = \Theta^+$, $p = p^+$ and $L = L+1$. At the moment of splitting, the neighborhood configuration at a given site $s \in \mathscr{S}_\lambda$ is unknown. Thus the calculation of the term $P(\omega^+ \mid \widehat{\beta}, p^+, L+1)$ is not possible. First, we have to provide a tentative labeling of the sites in $\mathscr{S}_\lambda$. Then we can sample the posterior distribution using a Gibbs sampler. Of course, a tentative labeling might be obtained by allocating λ_1 and λ_2 at random. In practice, however, we need a labeling ω^+ which has a relatively high posterior probability in order to maintain a reasonable acceptance probability. To achieve this goal, we use a few step (around 5 iterations) of ICM [2] algorithm to obtain a suboptimal segmentation of $\mathscr{S}_\lambda$. The resulting label map can be used to draw a sample from the posterior distribution using a one step Gibbs sampler [4]. The obtained ω^+ has a relatively high posterior probability since the tentative labeling was close to the optimal labeling.

3.2 Merging Two Classes

A pair (λ_1, λ_2) is chosen with a probability inversely proportional to their distance:

$$P_{select}^{merge}(\lambda_1, \lambda_2) = \frac{1/d(\lambda_1, \lambda_2)}{\sum\limits_{\lambda \in \Lambda} \sum\limits_{\kappa \in \Lambda} 1/d(\lambda, \kappa)} \tag{18}$$

where $d(\lambda_1, \lambda_2)$ is the combination of the Mahalanobis distance between the classes λ_1 and λ_2 defined as:

$$d(\lambda_1, \lambda_2) = (\vec{\mu}_{\lambda_1} - \vec{\mu}_{\lambda_2})\, \Sigma_{\lambda_1}^{-1} (\vec{\mu}_{\lambda_1} - \vec{\mu}_{\lambda_2}) + (\vec{\mu}_{\lambda_2} - \vec{\mu}_{\lambda_1})\, \Sigma_{\lambda_2}^{-1} (\vec{\mu}_{\lambda_2} - \vec{\mu}_{\lambda_1}) \tag{19}$$

In this way, we favor merging classes that are close to each other thus increasing acceptance probability. The merge proposal is deterministic, once the choices of λ_1 and λ_2 have been made. These two components are merged, reducing L by 1. As in the case of splitting, altering L changes the dimensionality of the variables Θ and p. The new parameter values (Θ^-, p^-) are obtained from Eq. (9)–(11). The reallocation is simply done by setting the label at sites $s \in \mathscr{S}_{\{\lambda_1,\lambda_2\}}$ to the new label λ. The random variables u are obtained by back-substitution into Eq. (13)–(17).

3.3 Acceptance Probability

In the Metropolis-Hastings algorithm [6], the split or merge proposal is accepted with a probability relative to the probability ratio of the current and the proposed states. Let us first consider the acceptance probability $\mathscr{A}_{split}$ for the split move. For the corresponding merge move, the acceptance probability is obtained as the inverse of the same expression with some obvious differences in the substitutions. $\mathscr{A}_{split}(L,\widehat{p},\widehat{\beta},\widehat{\omega},\widehat{\Theta};L+1,p^+,\widehat{\beta},\omega^+,\Theta^+) = \min(1,A)$, where A is given by:

$$A = \frac{P(L+1,p^+,\widehat{\beta},\omega^+,\Theta^+ \mid \mathscr{F})}{P(L,\widehat{p},\widehat{\beta},\widehat{\omega},\widehat{\Theta} \mid \mathscr{F})} \frac{P_{merge}(L+1)P^{merge}_{select}(\lambda_1,\lambda_2)}{P_{split}(L)P^{split}_{select}(\lambda)P_{realloc}}$$
$$\times \quad \frac{1}{P(u1)\prod_{i=1}^{3}\left(P(u2_i)\prod_{j=i}^{3}P(u3_{i,j})\right)} \left|\frac{\partial \psi}{\partial(\Theta_\lambda,p_\lambda,u)}\right|$$

$P_{realloc}$ denotes the probability of reallocating pixels labeled by λ into regions labeled by λ_1 and λ_2. It can be derived from Eq. (7) by restricting the set of labels Λ^+ to the subset $\{\lambda_1,\lambda_2\}$ and taking into account only those sites s for which $\omega^+_s \in \{\lambda_1,\lambda_2\}$. The last factor is the Jacobian determinant of the transformation ψ:

$$\left|\frac{\partial \psi}{\partial(\Theta_\lambda,p_\lambda,u)}\right| = -p_\lambda \prod_{i=1}^{3}\left(\frac{\Sigma^2_{i,i}}{u1\,(u1-1)}\left(1-u2_i{}^2\right)\left(1-u3_{i,i}\right) \times u3_{i,i}\prod_{j=i}^{3}\frac{\Sigma_{i,j}}{u1\,(u1-1)}\right)$$

The acceptance probability for the merge move can now be easily derived with some obvious differences in the substitutions as $\mathscr{A}_{merge}(L,\widehat{p},\widehat{\beta},\widehat{\omega},\widehat{\Theta};L-1,p^-,\widehat{\beta},\omega^-,\Theta^-) = \min\left(1,\frac{1}{A}\right)$

4 Optimization According to the MAP Criterion

The following MAP estimator is used to obtain an optimal segmentation $\widehat{\omega}$ and model parameters $\widehat{L},\widehat{p},\widehat{\beta},\widehat{\Theta}$:

$$(\widehat{\omega},\widehat{L},\widehat{p},\widehat{\beta},\widehat{\Theta}) = \arg\max_{L,p,\beta,\omega,\Theta} P(L,p,\beta,\omega,\Theta \mid \mathscr{F}) \tag{20}$$

with the following constraints: $\omega \in \Omega, L_{min} \leq L \leq L_{max}, \sum_{\lambda\in\Lambda} p_\lambda = 1, \forall\lambda \in \Lambda : 0 \leq \mu_{\lambda,i} \leq 1, 0 \leq \Sigma_{\lambda,i,i} \leq 1$, and $-1 \leq \Sigma_{\lambda,i,j} \leq 1$. Eq. (20) is a combinatorial optimization problem which can be solved using simulated annealing [4]:

Algorithm 1 (RJMCMC Segmentation)

① *Set $k = 0$, and initialize $\widehat{\beta}^0$, $\widehat{L}^0$, $\widehat{p}^0$, $\widehat{\Theta}^0$, and the initial temperature T_0.*

Original image | Segmentation result (10 classes)

Figure 2: Segmentation of image *kodakBus93*.

② *A sample* $(\widehat{\omega}^k, \widehat{L}^k, \widehat{p}^k, \widehat{\beta}^k, \widehat{\Theta}^k)$ *is drawn from the posterior distribution using the* hybrid sampler *outlined in Section 3. Each sub-chain is sampled via the corresponding move-type while all the other parameter values are set to their current estimate.*

③ *Goto Step* ② *with* $k = k+1$ *and* T_{k+1} *until* $k < \mathcal{K}$*.*

As usual, an exponential annealing schedule ($T_{k+1} = 0.98T_k$, $T_0 = 6.0$) was chosen so that the algorithm would converge after a reasonable number of iterations. In our experiments, the algorithm was stopped after 200 iterations ($T_{200} \approx 0.1$).

Although our discussion includes the estimation of the MRF hyperparameter β, we haven't estimated it in our experiments. Since the likelihood is approximated by the pseudo-likelihood, the posterior density of β may not be proper under particular label configurations. This problem has been reported in [1] and our findings were similar. Fortunately, the mixture weights p_λ are able to maintain a balance between the external and internal field strength, which makes it possible to set β a priori.

5 Experimental Results

The proposed algorithm has been tested on a variety of real color images. Herein, we present a few examples of these results. First, the original images were converted from RGB to LHS color space in which chroma and intensity informations are separated. The dynamic range of all color components was normalized such that they take their values from $(0,1)$. Independently of the input image, we start the algorithm with two classes ($\widehat{L}^0 = 2$), each of them having equal weights ($\widehat{p}_0^0 = \widehat{p}_1^0 = 0.5$). The mean vectors were set to $[0.2, 0.2, 0.2]$ and $[0.7, 0.7, 0.7]$. The diagonal elements of the covariance matrices were set to 0.05, while other elements were set to 0.00001. The hyperparameter β was fixed a priori. We found that $\beta = 2.5$ gives good results on all of the tested images. Furthermore, the number of classes L was restricted to the interval $[1, 50]$.

In Fig. 1, 3 classes have been identified and an accurate segmentation is obtained. A more difficult scene is segmented in Fig. 2, where 10 classes have been detected. In Fig. 3, a background-foreground style segmentation is shown. The test images can be found at the Kodak Digital Image Offering WWW site: `http://www.kodak.com/digitalImaging/samples/imageIntro.shtml`.

Original image   Segmentation result (9 classes)

Figure 3: Segmentation of image *bird11*.

We found that the acceptance rate for the split or merge move was $\approx 5\%$ which is quite reasonable considering the fact that this move type involves a change of the number of classes only. A similar finding has also been reported in [14].

6 Conclusion

In this paper, we have proposed a new RJMCMC sampler for multivariate Gaussian mixture identification and applied it to unsupervised color image segmentation. For this purpose, we have established a Bayesian segmentation model using MRF modeling of the underlying label field. Pixel classes are represented by multivariate Gaussian distributions. The number of classes, class model parameters, and pixel labels are all directly sampled from the posterior distribution using our RJMCMC sampler. A single parameter is defined a priori which defines the interaction strength of neighboring pixels. The final estimates, satisfying the MAP criterion, are obtained through simulated annealing. Experimental results show that an accurate segmentation can be obtained on a variety of real images.

References

[1] S. A. Barker and P. J. W. Rayner. Unsupervised image segmentation using Markov random field models. *Pattern Recognition*, 33(4):587–602, Apr 2000.

[2] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, series B*, 48(3):259–302, 1986.

[3] D. Cremers, F. Tischhauser, J. Weickert, and C. Schnorr. Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.

[4] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[5] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–731, 1995.

[6] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their application. *Biometrika*, 57:97–109, 1970.

[7] C. L. Huang, T. Y. Cheng, and C. C. Chen. Color images segmentation using scale space filter and Markov random field. *Pattern Recognition*, 25(10):1217–1229, 1992.

[8] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(8):799–813, August 1989.

[9] D. A. Langan, J. W. Modestino, and J. Zhang. Cluster validation for unsupervised stochastic model-based image segmentation. *IEEE Trans. on Image Processing*, 7(2):180–195, February 1998.

[10] J. Liu and Y. H. Yang. Multiresolution color image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(7):689–700, July 1994.

[11] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, Duluth, London, 1979.

[12] D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of textured color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(10):939–954, October 1995.

[13] S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, series B*, 59(4):731–792, 1997.

[14] C. P. Robert, T. Rydén, and D. M. Titterington. Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method. *Journal of the Royal Statistical Society, series B*, 62(1):57–75, 2000.

[15] C. S. Won and H. Derin. Unsupervised segmentation of noisy and textured images using Markov random fields. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 54(4):308–328, July 1992.

[16] Song Chun Zhu. Stochastic jump-diffusion process for computing medial axes in Markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(11):1158–1169, November 1999.