Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 26 (2008) 361–371
www.elsevier.com/locate/imavis

# Segmentation of color images via reversible jump MCMC sampling

Zoltan Kato *

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

Received 25 February 2005; received in revised form 5 June 2006; accepted 8 December 2006

**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. The algorithm has been validated on a database of real images with human segmented ground truth.


*Keywords:* Unsupervised image segmentation; Color; Parameter estimation; Normal mixture identification; Markov random fields; Reversible jump Markov chain Monte Carlo; Simulated annealing

## 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 [1,2], 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 [3]. 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* [4,5]. 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 [6,7]: segmentations and parameter estimates are obtained via an iterative algorithm by alternately sampling the label field based on the current estimates of the parameters.

* Tel.: +36 62 546 399; fax: +36 62 546 397.
*E-mail address:* kato@inf.u-szeged.hu
*URL:* http://www.inf.u-szeged.hu/~kato/

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.12.004

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 [1,8]: 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 [9], 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 [9]. Until now, RJMCMC has been applied to univariate Gaussian mixture identification [10] and its applications in different areas like inference in hidden Markov models [11], intensity-based image segmentation [12], and computing medial axes of 2D shapes [13]. The novelty of our approach is twofold: first, we extend the ideas in [10,12] and 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, it is straightforward to extend the equations to higher dimensions.

## 2. Color image segmentation model

The model assumes that the real world scene consists of a set of regions whose observed color changes slowly, but across the boundary between them, they change abruptly. What we want to infer is a *labeling* $\omega$ consisting of a simplified, abstract version of the input image: regions has a constant value (called a *label* in our context) and the discontinuities between them form a curve – the contour. Such a labeling $\omega$ specifies a *segmentation*. Taking the probabilistic approach, one usually wants to come up with a *probability measure* on the set $\Omega$ of all possible segmentations of the input image and then select the one with the highest probability. Note that $\Omega$ is finite, although huge. A widely accepted standard, also motivated by the human visual system [14,15], is to construct this probability measure in a Bayesian framework [16–18]. We will assume that we have a set of observed ($Y$) and hidden ($X$) random variables. In our context, the *observation* $\mathscr{F} \in Y$ represents the color values used for partitioning the image, and the hidden entity $\omega \in X$ represents the segmentation itself. Hence 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 $\vec{f}_s$. Note that color components are normalized. Furthermore, a segmentation $\omega$ assigns a label $\omega_s$ from the set of labels $\Lambda = \{1, 2, \ldots, L\}$ to each site $s$.

First, we have to quantify how well any occurrence of $\omega$ fits $\mathscr{F}$. This is expressed by the probability distribution $P(\mathscr{F}|\omega)$ – the *imaging model*. Second, we define a set of properties that any segmentation $\omega$ must posses regardless the image data. These are described by $P(\omega)$, the *prior*, which tells us how well any occurrence $\omega$ satisfies these properties. For that purpose, $\omega_s$ is modeled as a discrete random variable taking values in $\Lambda$. The set of these labels $\omega = \{\omega_s, s \in \mathscr{S}\}$ is a random field, called the *label process*. Furthermore, the observed color features are supposed to be a realization $\mathscr{F}$ from another random field, which is a function of the label process $\omega$. Basically, the *image process* $\mathscr{F}$ represents the manifestation of the underlying label process. The multivariate Normal density is typically an appropriate model for such classification problems where the feature vectors $\vec{f}_s$ for a given class $\lambda$ are mildly corrupted versions of a single mean vector $\mu_\lambda$ [19,20]. Applying these ideas, the *image process* $\mathscr{F}$ can be formalized as follows: $P(\vec{f}_s|\omega_s)$ follows a three-variate Gaussian distribution $N(\vec{\mu}, \mathbf{\Sigma})$, each pixel class $\lambda \in \Lambda = \{1, 2, \ldots, L\}$ is represented by its mean vector $\vec{\mu}_\lambda$ and covariance matrix $\mathbf{\Sigma}_\lambda$. As for the *label process* $\omega$, a MRF model is adopted [21] over a nearest neighborhood system. According to the *Hammersley–Clifford theorem* [9], $P(\omega)$ follows a Gibbs distribution:

$$P(\omega) = \frac{1}{Z}\exp(-U(\omega)) = \frac{1}{Z}\exp\left(-\sum_{C\in\mathscr{C}} V_C(\omega_C)\right), \tag{1}$$

where $U(\omega)$ is called an *energy function*, $Z = \sum_{\omega\in\Omega}\exp(-U(\omega))$ is the normalizing constant (or *partition function*) and $V_C$ denotes the *clique potentials* of cliques $C \in \mathscr{C}$ having the label configuration $\omega_C$. The prior $P(\omega)$ will represent the simple fact that segmentations should be locally homogeneous. Therefore we will define clique potentials $V_C$ over pairs of neighboring pixels *(doubletons)* such that similar classes in neighboring pixels are favored

$$V_C = \beta \cdot \delta(\omega_s, \omega_r) = \begin{cases} +\beta & \text{if } \omega_s \neq \omega_r, \\ -\beta & \text{otherwise,} \end{cases} \tag{2}$$

where $\beta$ is a hyper-parameter controlling the interaction strength. As $\beta$ increases, regions become more homogeneous. The energy is proportional to the length of the region boundaries. Thus homogeneous segmentations will get a higher probability, as expected.

Factoring the above distributions and applying the Bayes theorem gives us the *posterior* distribution $P(\omega|\mathscr{F}) \propto P(\mathscr{F}|\omega)P(\omega)$. Note that the constant factor $1/P(\mathscr{F})$ has been dropped as we are only interested in $\hat{\omega}$

which *maximizes* the posterior, i.e. the Maximum A Posteriori (MAP) estimate of the hidden field $X$

$$\hat{\omega} = \arg\max_{\omega\in\Omega} P(\mathcal{F}|\omega)P(\omega). \tag{3}$$

The models of the above distributions depend also on certain parameters. Since neither these parameters nor $X$ is known, both has to be inferred from the only observable entity $Y$. This is known in statistics as the *incomplete data* problem and a fairly standard tool to solve it is *Expectation Maximization* [22] and its variants. However, our problem becomes much harder when the number of labels $L$ is unknown. When this parameter is also being estimated, the unsupervised segmentation problem may be treated as a *model selection* problem over a combined model space. From this point of view, $L$ becomes a *model indicator* and the observation $\mathcal{F}$ is regarded as a three-variate Normal *mixture* with $L$ components corresponding to clusters of pixels which are homogeneous in color.

The goal of our analysis is inference about the number $L$ of Gaussian mixture components (each one corresponds to a label), the component parameters $\Theta = \{\Theta_\lambda = (\vec{\mu}_\lambda, \Sigma_\lambda)|\lambda \in \Lambda\}$, the component weights $p_\lambda$ summing to 1, the inter-pixel interaction strength $\beta$, and the segmentation $\omega$. The only observable entity is $\mathcal{F}$, thus the posterior distribution becomes

$$P(L,p,\beta,\omega,\Theta|\mathcal{F}) = P(L,p,\beta,\omega,\Theta,\mathcal{F})/P(\mathcal{F}). \tag{4}$$

Note that $P(\mathcal{F})$ is constant, hence we are only interested in the joint distribution of the variables $L$, $p$, $\beta$, $\Theta$, $\mathcal{F}$

$$P(L,p,\beta,\omega,\Theta,\mathcal{F}) = P(\omega,\mathcal{F}|\Theta,\beta,p,L)P(\Theta,\beta,p,L). \tag{5}$$

In our context, it is natural to impose conditional independences on $(\Theta,\beta,p,L)$ so that their joint probability reduces to the product of priors

$$P(\Theta,\beta,p,L) = P(\Theta)P(\beta)P(p)P(L). \tag{6}$$

Let us concentrate now on the posterior of $(\omega, \mathcal{F})$

$$P(\omega,\mathcal{F}|\Theta,\beta,p,L) = P(\mathcal{F}|\omega,\Theta,\beta,p,L)P(\omega|\Theta,\beta,p,L). \tag{7}$$

Before further proceeding, we can impose additional conditional independences. Since each pixel class (or label) is represented by a Gaussian, we obtain

$$\begin{aligned} P(\mathcal{F}|\omega,\Theta,\beta,p,L) &= P(\mathcal{F}|\omega,\Theta) \\ &= \prod_{s\in\mathcal{S}} \frac{1}{\sqrt{(2\pi)^3|\Sigma_{\omega_s}|}} \exp\left(-\frac{1}{2}\left(\vec{f}_s - \vec{\mu}_{\omega_s}\right)\right. \\ &\quad \left.\times \Sigma_{\omega_s}^{-1}\left(\vec{f}_s - \vec{\mu}_{\omega_s}\right)^{\mathrm{T}}\right) \end{aligned} \tag{8}$$

and $$P(\omega|\Theta,\beta,p,L) = P(\omega|\beta,p,L). \tag{9}$$

Furthermore, the component weights $p_\lambda$, $\lambda \in \Lambda$ can be incorporated into the underlying MRF label process as an external field strength. Formally this is done via the *singleton* potential (probability of individual pixel labels)

$$P(\omega|\beta,p,L) = P(\omega|\beta,L)\prod_{s\in\mathcal{S}} p_{\omega_s}. \tag{10}$$

Since the label process follows a Gibbs distribution [9], we can also express the above probability in terms of an energy

$$P(\omega|\beta,p,L) = \frac{1}{Z(\beta,p,L)}\exp(-U(\omega|\beta,p,L)), \quad \text{where} \tag{11}$$

$$U(\omega|\beta,p,L) = \beta\sum_{\{s,r\}\in\mathcal{C}}\delta(\omega_s,\omega_r) - \sum_{s\in\mathcal{S}}\log(p_{\omega_s}) \tag{12}$$

$\{s,r\}$ denotes a *doubleton* containing the neighboring pixel sites $s$ and $r$. The basic idea is that segmentations has to be homogeneous *and* only those labels are valid in the model for which we can associate fairly big regions. The former constraint is ensured by the doubletons while the latter one is implemented via the component weights. Indeed, invalid pixel classes typically get only a few pixels assigned hence no matter how homogeneous are the corresponding regions, the above probability will be low. Unfortunately, the partition function $Z(\beta,p,L)$ is not tractable [23], thus the comparison of the likelihood of two differing MRF realizations from Eq. (11) is infeasible. Instead, we can compare their Pseudo-Likelihood [23]

$$P(\omega|\beta,p,L) \approx \prod_{s\in\mathcal{S}} \frac{p_{\omega_s}\exp\left(-\beta\sum_{\forall r:\{s,r\}\in\mathcal{C}}\delta(\omega_s,\omega_r)\right)}{\sum_{\lambda\in\Lambda}p_\lambda\exp\left(-\beta\sum_{\forall r:\{s,r\}\in\mathcal{C}}\delta(\lambda,\omega_r)\right)} \tag{13}$$

Finally, we get the following approximation for the whole posterior distribution:

$$\begin{aligned} P(L,p,\beta,\omega,\Theta|\mathcal{F}) \propto\ & P(\mathcal{F}|\omega,\Theta)\prod_{s\in\mathcal{S}} \frac{p_{\omega_s}\exp\left(-\beta\sum_{\forall r:\{s,r\}\in\mathcal{C}}\delta(\omega_s,\omega_r)\right)}{\sum_{\lambda\in\Lambda}p_\lambda\exp\left(-\beta\sum_{\forall r:\{s,r\}\in\mathcal{C}}\delta(\lambda,\omega_r)\right)} \\ & \times P(\beta)P(L) \\ & \times \prod_{\lambda\in\Lambda} P(\vec{\mu}_\lambda)P(\Sigma_\lambda)P(p_\lambda). \end{aligned} \tag{14}$$

In order to simplify our presentation, we will follow [10] 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. Although it is theoretically possible to sample $\beta$ from the posterior, we will set its value *a priori*. The reasons are as follows:

- Due to the approximation by the Pseudo-Likelihood, the posterior density for $\beta$ may not be proper [12].
- Being a hyper-parameter, $\beta$ is largely independent of the input image. As long as it is large enough, the quality of segmentations are quite similar [21]. In addition, it is also independent of the number of classes since doubleton potentials will only check whether two neighboring labels are equal.

As a consequence, $P(\beta)$ is constant.

## 3. Sampling from the posterior distribution

A broadly used tool to sample from the posterior distribution in Eq. (14) is the Metropolis–Hastings method [24]. Classical methods, however, cannot 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 [3]. When we have multiple parameter subspaces of different dimensionality, it is necessary to devise different *move types* between the subspaces [3]. These will be combined in a so called *hybrid sampler*. For the color image segmentation model, the following move types are needed:

(1) sampling the labels $\omega$ (i.e. 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 $\beta$;
(5) sampling the number of classes $L$ (splitting one mixture component into two, or combining two into one).

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. (14) by setting unaffected parameters to their current estimate. For example, in move (1), the parameters $L, p, \beta, \Theta$ are set to their estimates $\hat{L}, \hat{p}, \hat{\beta}, \hat{\Theta}$. Thus the posterior in Eq. (14) reduces to the following form:

$$\begin{aligned} P(L,p,\beta,\omega,\Theta|\mathscr{F}) &\propto P(\mathscr{F}|\omega,\hat{\Theta})P(\omega|\hat{\beta},\hat{p},\hat{L}) \\ &\propto \prod_{s\in\mathscr{S}} \left( \frac{1}{\sqrt{(2\pi)^3|\hat{\Sigma}_{\omega_s}|}} \exp\left( -\frac{1}{2}\left(\vec{f}_s - \vec{\hat{\mu}}_{\omega_s}\right) \right.\right. \\ &\quad \left.\left. \times \hat{\Sigma}_{\omega_s}^{-1}\left(\vec{f}_s - \vec{\hat{\mu}}_{\omega_s}\right)^{\mathrm{T}} \right) \right) \\ &\quad \times \prod_{s\in\mathscr{S}} \hat{p}_{\omega_s} \exp\left( -\hat{\beta} \sum_{\forall r:\{s,r\}\in\mathscr{C}} \delta(\omega_s,\omega_r) \right). \end{aligned} \tag{15}$$

Basically, the above equation corresponds to a segmentation with *known parameters*.

In our experiments, move (4) is never executed since $\beta$ is fixed a priori. As for moves (2) and (3), a closed form solution also exists: Using the current label field $\hat{\omega}$ as a *training set*, an unbiased estimate of $p_\lambda, \vec{\mu}_\lambda$, and $\Sigma_\lambda$ can be obtained as the zeroth, first and second moments of the labeled data [20,21]. We have used this approach in our experiments.

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 $\omega$, $\Theta$ and $p$.

### 3.1. Reversible jump mechanism

First, let us briefly review the reversible jump technique. A comprehensive introduction by Green can be found in [25]. For ease of notation, we will denote the set of unknowns $\{L, p, \beta, \omega, \Theta\}$ by $\chi$ and let $\pi(\chi)$ be the target probability measure (the posterior distribution from Eq. (14), in our context). A standard tool to sample from $\pi(\chi)$ is the Metropolis–Hastings method [26,24]: assuming the current state is $\chi$

(1) first a candidate new state is drawn from the *proposal measure* $q(\chi, \chi')$, which is an essentially arbitrary joint distribution. Often a uniform distribution is adopted in practice.
(2) Then $\chi'$ is accepted with probability $\mathscr{A}(\chi, \chi')$ – the so called *acceptance probability*.

If $\chi'$ is rejected then we stay in the current state $\chi$. Otherwise a transition $\chi \to \chi'$ is made. The sequence of accepted states is a Markov chain. As usual in MCMC [25], this chain has to be reversible which implies that the transition kernel $P$ of the chain satisfies the *detailed balance* condition

$$\int \pi(\mathrm{d}\chi) P(\chi, \mathrm{d}\chi') = \int \pi(\mathrm{d}\chi') P(\chi', \mathrm{d}\chi). \tag{16}$$

From the above equation, $\mathscr{A}(\chi, \chi')$ can be formally derived [24,25]

$$\mathscr{A}(\chi, \chi') = \min\left(1, \frac{\pi(\chi')q(\chi, \chi')}{\pi(\chi)q(\chi', \chi)}\right). \tag{17}$$

The implementation of these transitions are quite straightforward. Following Green [25], we can easily separate the random and deterministic parts of such a transition in the following manner:

- At the current state $\chi$, we generate a random vector $u$ of dimension $r$ from a known density $p$. Then the candidate new state is formed as a deterministic function of the current state $\chi$ and the random numbers in $u$: $\chi' = h(\chi, u)$.
- Similarly, the reverse transition $\chi' \to \chi$ would be accomplished with the aid of $r'$ random numbers $u'$ drawn from $p'$, yielding $\chi = h'(\chi', u')$.

If the transformation from $(\chi, u)$ to $(\chi', u')$ is a *diffeomorphism* (i.e. both the transformation and its inverse are differentiable), then the *detailed balance* condition is satisfied when [25]

$$\pi(\chi)p(u)\mathscr{A}(\chi, \chi') = \pi(\chi')p'(u')\mathscr{A}(\chi', \chi)\left|\frac{\partial(\chi', u')}{\partial(\chi, u)}\right|, \tag{18}$$

where the last factor is the Jacobian of the diffeomorphism. Note that it appears in the equality only because the pro-

posal destination $\chi' = h(\chi, u)$ is specified indirectly. The *acceptance probability* is derived again from the *detailed balance* equation [3,25]

$$\mathscr{A}(\chi, \chi') = \min\left(1, \frac{\pi(\chi')p'(u')}{\pi(\chi)p(u)}\left|\frac{\partial(\chi', u')}{\partial(\chi, u)}\right|\right). \tag{19}$$

The main advantage of the above formulation is that it remains valid in a variable dimension context. As long as the transformation $(\chi, u) \mapsto (\chi', u')$ remains a *diffeomorphism*, the dimensions of $\chi$ and $\chi'$ (denoted by $d$ and $d'$) can be different. One necessary condition for that is the so called *dimension matching* (see Fig. 1). Indeed, if the $d + r = d' + r'$ equality failed then the mapping and its inverse could not both be differentiable.

In spite of the relatively straightforward theory of reversible jumps, it is by far not evident how to construct efficient jump proposals in practice. This is particularly true in image processing problems, where the dimension of certain inferred variables (like the labeling $\omega$) is quite big. Although there have been some attempt [27,25] to come up with general recipes on how to construct efficient proposals, there is still no good solution to this problem.

In the remaining part of this section, we will apply the reversible jump technique for sampling from the posterior in Eq. (14). In particular, we will construct a diffeomorphism $\psi$ along with the necessary probability distributions of the random variables $u$ such that a reasonable acceptance rate of jump proposals is achieved. In our case, a jump proposal may either be a *split* or *merge* of classes. In order to implement these proposals, we will extend the moment matching concept of Green [3,10] to three-variate Gaussians. However, our construction is admittedly ad hoc and fine-tuned to the color image segmentation problem. For a theoretical treatment of the multi-variate Gaussian case, see the works of Stephens [28,29].

### 3.2. Splitting one class into two

The *split proposal* begins by randomly choosing a class $\lambda$ with a uniform probability $P^{\text{split}}_{\text{select}}(\lambda) = 1/L$. Then $L$ is increased by 1 and $\lambda$ is split into $\lambda_1$ and $\lambda_2$. In doing so, a new set of parameters need to be generated. Altering $L$ changes the dimensionality of the variables $\Theta$ and $p$. Thus we shall define a deterministic function $\psi$ as a function of these Gaussian mixture parameters

$$(\Theta^+, p^+) = \psi(\Theta, p, u), \tag{20}$$

where the superscript + denotes parameter vectors after incrementing $L$.$u$ is a set of random variables having as many elements as the degree of freedom of joint variation of the current parameters $(\Theta, p)$ and the proposal $(\Theta^+, p^+)$. Note that this definition satisfies the *dimension matching* constraint [3] (see Fig. 1), which guarantees that one can jump back and forth between different parameter sub-spaces. The new parameters of $\lambda_1$ and $\lambda_2$ are assigned by matching the 0th, 1th, 2th moments of the component being split to those of a combination of the two new components [10]:

$$p_\lambda = p^+_{\lambda_1} + p^+_{\lambda_2}, \tag{21}$$

$$p_\lambda \vec{\boldsymbol{\mu}}_\lambda = p^+_{\lambda_1} \vec{\boldsymbol{\mu}}^+_{\lambda_1} + p^+_{\lambda_2} \vec{\boldsymbol{\mu}}^+_{\lambda_2}, \tag{22}$$

$$p_\lambda\left(\vec{\boldsymbol{\mu}}_\lambda \vec{\boldsymbol{\mu}}^{\mathrm{T}}_\lambda + \boldsymbol{\Sigma}_\lambda\right) = p^+_{\lambda_1}\left(\vec{\boldsymbol{\mu}}^+_{\lambda_1} \vec{\boldsymbol{\mu}}^{+\mathrm{T}}_{\lambda_1} + \boldsymbol{\Sigma}^+_{\lambda_1}\right) + p^+_{\lambda_2}\left(\vec{\boldsymbol{\mu}}^+_{\lambda_2} \vec{\boldsymbol{\mu}}^{+\mathrm{T}}_{\lambda_2} + \boldsymbol{\Sigma}^+_{\lambda_2}\right). \tag{23}$$

There are 10 degrees of freedom in splitting $\lambda$ since covariance matrices are symmetric. Therefore, we need to generate a random variable $u1$, a random vector $\vec{\boldsymbol{u2}}$ and a symmetric random matrix $\boldsymbol{u3}$. We can now define the diffeomorphism $\psi$ which transforms the old parameters $(\Theta, \mathrm{p})$ to the new $(\Theta^+, p^+)$ using the above moment equations and the random numbers $u1$, $\vec{\boldsymbol{u2}}$ and $\boldsymbol{u3}$:

$$p^+_{\lambda_1} = p_\lambda u1 \tag{24}$$

$$p^+_{\lambda_2} = p_\lambda(1 - u1) \tag{25}$$

$$\mu^+_{\lambda_1, i} = \mu_{\lambda, i} + u2_i\sqrt{\Sigma_{\lambda,i,i}\frac{1 - u1}{u1}} \tag{26}$$

$$\mu^+_{\lambda_2, i} = \mu_{\lambda, i} - u2_i\sqrt{\Sigma_{\lambda,i,i}\frac{u1}{1 - u1}} \tag{27}$$

$$\Sigma^+_{\lambda_1,i,j} = \begin{cases} u3_{i,i}\left(1 - u2_i^2\right)\Sigma_{\lambda,i,i}\frac{1}{u1} & \text{if } i = j \\ u3_{i,j}\Sigma_{\lambda,i,j}\sqrt{\left(1 - u2_i^2\right)}\sqrt{\left(1 - u2_j^2\right)u3_{i,i}u3_{j,j}} & \text{if } i \neq j \end{cases} \tag{28}$$

$$\Sigma^+_{\lambda_2,i,j} = \begin{cases} (1 - u3_{i,j})\left(1 - u2_i^2\right)\Sigma_{\lambda,i,i}\frac{1}{u1} & \text{if } i = j \\ (1 - u3_{i,j})\Sigma_{\lambda,i,i}\sqrt{\left(1 - u2_i^2\right)\left(1 - u2_j^2\right)} \\ \quad \times\sqrt{(1 - u3_{i,i})(1 - u3_{j,j})} & \text{if } i \neq j \end{cases} \tag{29}$$

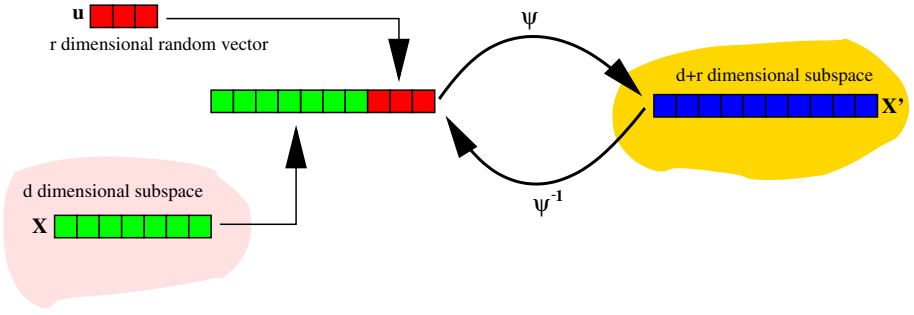


Fig. 1. $\psi$ is a *diffeomorphism* which transforms back and forth between parameter subspaces of different dimensionality. *Dimension matching* can be implemented by generating a random vector $u$ such that the dimensions of $(X, u)$ and $X'$ are equal.

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 [30], or one can use the well-known Wishart distribution [31]. However, we did not experience such problems, mainly because the obtained covariance matrices are also reestimated from the image 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 $s \in \mathcal{S}_\lambda$ 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 $\omega^+$ is drawn from the posterior distribution with $\Theta = \Theta^+$, $p = p^+$ and $L = L + 1$. At the moment of splitting, however, the neighborhood configuration at a given site $s \in \mathcal{S}_\lambda$ is unknown. Thus the calculation of the term $P(\omega^+|\hat{\beta}, p^+, L+1)$ is not possible. First, we have to provide a tentative labeling of the sites in $S_\lambda$. Then we can sample the posterior distribution using a Gibbs sampler. Of course, a tentative labeling might be obtained by allocating $\lambda_1$ and $\lambda_2$ at random. In practice, however, we need a labeling $\omega^+$ 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 [32] algorithm to obtain a suboptimal initial segmentation of $\mathcal{S}_\lambda$. The resulting label map can then be used to draw a sample from the posterior distribution using a one step Gibbs sampler [9]. The obtained $\omega^+$ has a relatively high posterior probability since the tentative labeling was close to the optimal one.

### 3.3. Merging two classes

A pair $(\lambda_1, \lambda_2)$ is chosen with a probability inversely proportional to their distance

$$P^{\text{merge}}_{\text{select}}(\lambda_1, \lambda_2) = \frac{1/d(\lambda_1, \lambda_2)}{\sum_{\lambda\in\Lambda}\sum_{\kappa\in\Lambda} 1/d(\lambda, \kappa)} \tag{30}$$

where $d(\lambda_1, \lambda_2)$ is the symmetric Mahalanobis distance between the classes $\lambda_1$ and $\lambda_2$ defined as

$$\begin{aligned} d(\lambda_1, \lambda_2) = {} & (\vec{\boldsymbol{\mu}}_{\lambda_1} - \vec{\boldsymbol{\mu}}_{\lambda_2})\boldsymbol{\Sigma}^{-1}_{\lambda_1}(\vec{\boldsymbol{\mu}}_{\lambda_1} - \vec{\boldsymbol{\mu}}_{\lambda_2}) \\ & + (\vec{\boldsymbol{\mu}}_{\lambda_2} - \vec{\boldsymbol{\mu}}_{\lambda_1})\boldsymbol{\Sigma}^{-1}_{\lambda_2}(\vec{\boldsymbol{\mu}}_{\lambda_2} - \vec{\boldsymbol{\mu}}_{\lambda_1}) \end{aligned} \tag{31}$$

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 $\lambda_1$ and $\lambda_2$ 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 $\Theta$ and $p$. The new parameter values $(\Theta^-, p^-)$ are obtained from Eqs. (21)–(23). The reallocation is simply done by setting the label at sites $s \in \mathcal{S}_{\{\lambda_1,\lambda_2\}}$ to the new label $\lambda$. The random variables $u$ are obtained by back-substitution into Eq. (24)–(29).

### 3.4. Acceptance probability

As discussed in Section 3.1, 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 $A_{\text{split}}$ 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.

$$\mathcal{A}_{\text{split}}(L, \hat{p}, \hat{\beta}, \hat{\omega}, \hat{\Theta}, L+1, p^+, \hat{\beta}, \omega^+, \Theta^+) = \min(1, A), \quad \text{where} \tag{32}$$

$$\begin{aligned} A = {} & \frac{P(L+1, p^+, \hat{\beta}, \omega^+, \Theta^+|\mathcal{F})}{P(L, \hat{p}, \hat{\beta}, \hat{\omega}, \hat{\Theta}|\mathcal{F})} \\ & \times \frac{P_{\text{merge}}(L+1)P^{\text{merge}}_{\text{select}}(\lambda_1, \lambda_2)}{P_{\text{split}}(L)P^{\text{split}}_{\text{select}}(\lambda)P_{\text{realloc}}} \\ & \times \frac{1}{P(u1)\prod_{i=1}^{3}\left(P(u2_i)\prod_{j=i}^{3}P(u3_{i,j})\right)}\left|\frac{\partial\psi}{\partial(\Theta_\lambda, p_\lambda, u)}\right| \end{aligned} \tag{33}$$

$P_{\text{realloc}}$ denotes the probability of reallocating pixels labeled by $\lambda$ into regions labeled by $\lambda_1$ and $\lambda_2$. It can be derived from Eq. (15) by restricting the set of labels $\Lambda^+$ to the subset $\{\lambda_1, \lambda_2\}$ and taking into account only those sites $s$ for which $\omega^+_s \in \{\lambda_1, \lambda_2\}$

$$\begin{aligned} P_{\text{realloc}} \approx {} & \prod_{\forall s:\omega^+_s\in\{\lambda_1,\lambda_2\}} \frac{1}{\sqrt{(2\pi)^3|\boldsymbol{\Sigma}^+_{\omega^+_s}|}}\exp\left(-\frac{1}{2}\left(\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}^+_{\omega^+_s}\right)\boldsymbol{\Sigma}^{+-1}_{\omega^+_s}\left(\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}^+_{\omega^+_s}\right)^T\right) \\ & \times \prod_{\forall s:\omega^+_s\in\{\lambda_1,\lambda_2\}} p^+_{\omega^+_s}\exp\left(-\hat{\beta}\sum_{\forall r:\{s,r\}\in\mathcal{C}}\delta(\omega^+_s, \omega^+_r)\right) \end{aligned} \tag{34}$$

The last factor is the Jacobian determinant of the transformation $\psi$

$$\begin{aligned} \left|\frac{\partial\psi}{\partial(\Theta_\lambda, p_\lambda, u)}\right| = {} & -p_\lambda\prod_{i=1}^{3}\left(\frac{\Sigma^2_{i,i}}{u1(u1-1)}(1-u2_i^2)(1-u3_{i,i})\times u3_{i,i}\right. \\ & \left.\times\prod_{j=i}^{3}\frac{\Sigma_{i,j}}{u1(u1-1)}\right) \end{aligned} \tag{35}$$

The acceptance probability for the merge move can be easily obtained with some obvious differences in the substitutions as

$$\begin{aligned} & \mathcal{A}_{\text{merge}}(L, \hat{p}, \hat{\beta}, \hat{\omega}, \hat{\Theta}; L-1, p^-, \hat{\beta}, \omega^-, \Theta^-) \\ & = \min\left(1, \frac{1}{A}\right) \end{aligned} \tag{36}$$

## 4. Optimization according to the MAP criterion

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

$$(\hat{\omega}, \hat{L}, \hat{p}, \hat{\beta}, \hat{\Theta}) = \arg \max_{L,p,\beta,\omega,\Theta} P(L, p, \beta, \omega, \Theta | \mathscr{F}) \quad (37)$$

with the following constraints: $\omega \in \Omega, L_{\min} \leqslant L \leqslant L_{\max}$, $\sum_{\lambda \in \Lambda} p_\lambda = 1, \forall \lambda \in \Lambda : 0 \leqslant \mu_{\lambda,i} \leqslant 1, 0 \leqslant \Sigma_{\lambda,i,i} \leqslant 1$, and $-1 \leqslant \Sigma_{\lambda,i,i} \leqslant 1$. Eq. (37) is a combinatorial optimization problem which can be solved using simulated annealing [9].

**Algorithm 1. (RJMCMC Segmentation)**

① Set $k = 0$. Initialize $\hat{\beta}^0, \hat{L}^0, \hat{p}^0, \hat{\Theta}^0$, and the initial temperature $T_0$.
② A sample $(\hat{\omega}^k, \hat{L}^k, \hat{p}^k, \hat{\beta}^k, \hat{\Theta}^k)$ 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 $k < \mathscr{K}$.

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

## 5. Experimental results

The evaluation of segmentation algorithms is inherently subjective. Nevertheless, there have been some recent work on defining an objective quality measure. Such a boundary benchmarking system is reported in [33] that we will use herein to quantify our results. The ground truth is provided as human segmented images (each image is processed by several subjects). The output of the benchmarked segmentation algorithm is presented to the system as a soft boundary map where higher values mean greater confidence in the existence of a boundary. Then two quantities are computed:

*Precision* is the probability that a machine-generated boundary pixel is a true boundary pixel. It measures the noisiness of the machine segmentation with respect to the human ones.

*Recall* is the probability that a true boundary pixel is detected. It tells us how much the ground truth is detected.

From these values, a precision-recall curve is produced which shows the tradeoff between the two quantities (see Fig. 6). We will also summarize the performance in a single number: the maximum *F-measure* value across an algorithm's precision-recall curve. The *F-measure* characterizes the distance of a curve from the origin which is computed as the harmonic mean of precision and recall [33]. Clearly, for non-intersecting precision-recall curves, the one with a higher maximum *F-measure* will dominate.

The proposed algorithm has been tested on a variety of real color images. First, the original images were converted

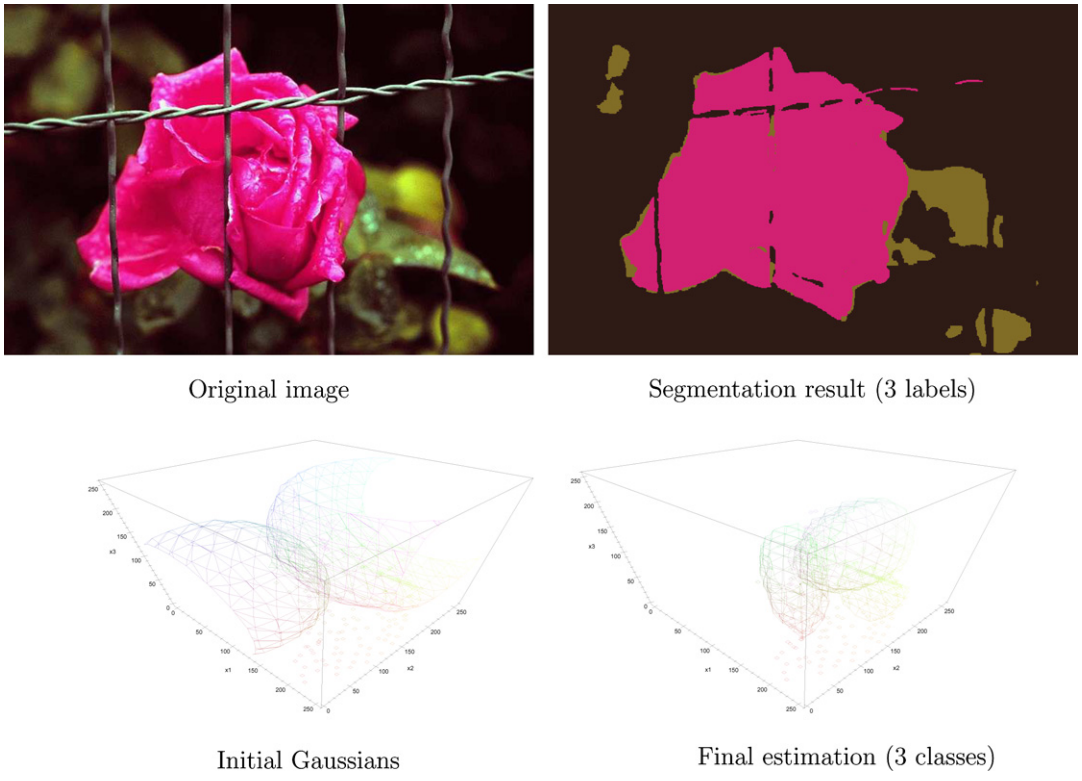


Fig. 2. Segmentation of image *rose41*.

from RGB to LHS color space [34] in which chroma and intensity informations are separated. Results in other color spaces can be found in [35]. The dynamic range of color components was then normalized to $(0,1)$. The number of classes $L$ was restricted to the interval $[1,50]$ and $\beta$ has been set to 2.5. This value gave us good results in all test

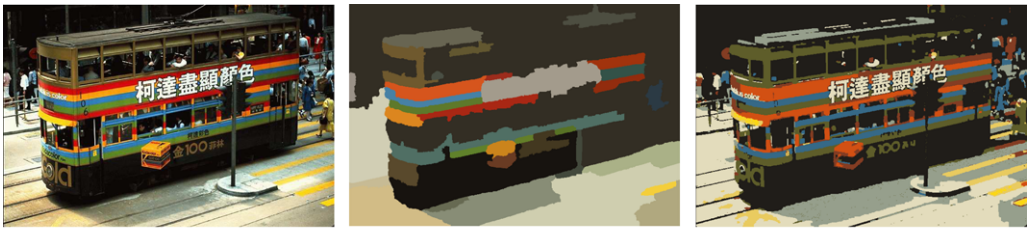


Fig. 3. Segmentation of image *kodakBus93*.

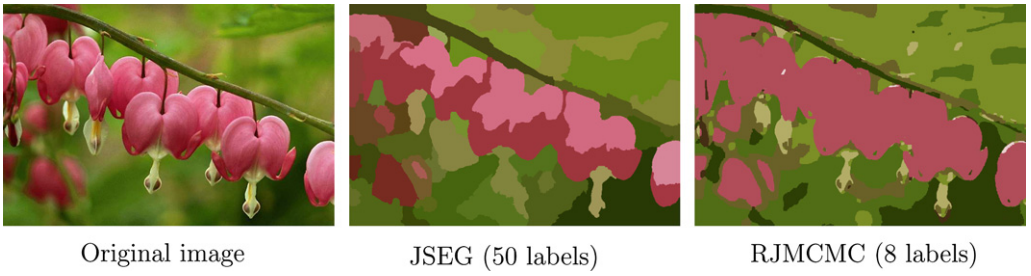


Fig. 4. Segmentation of image *bleedhearts34*.

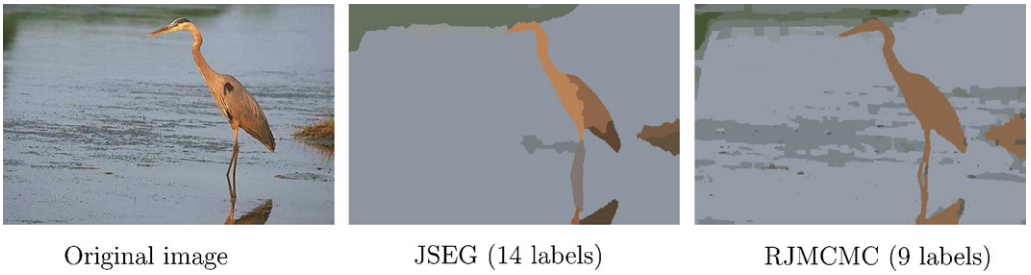


Fig. 5. Segmentation of image *bird11*.

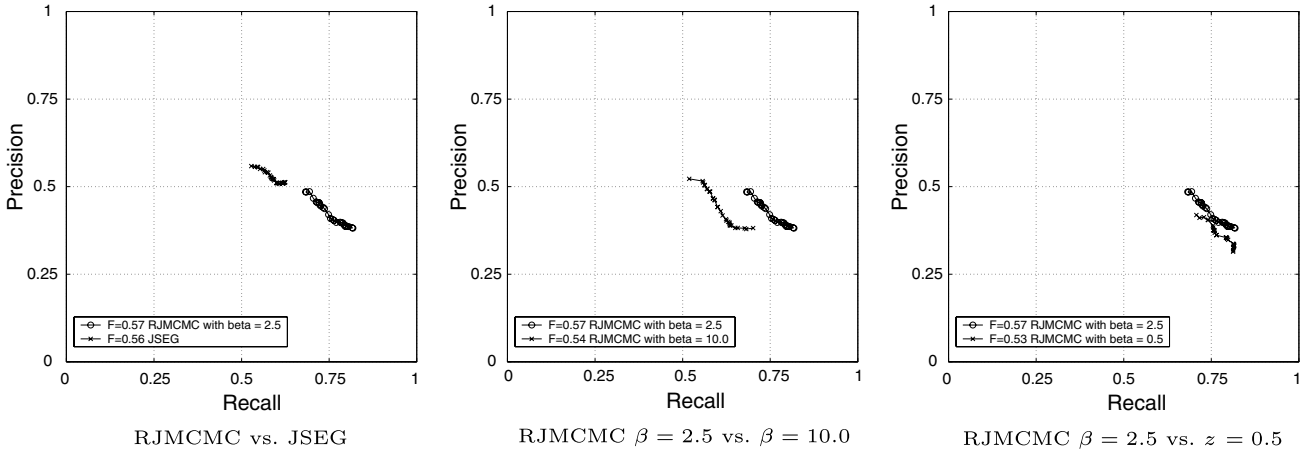


Fig. 6. Precision-recall curve for JSEG and RJMCMC.

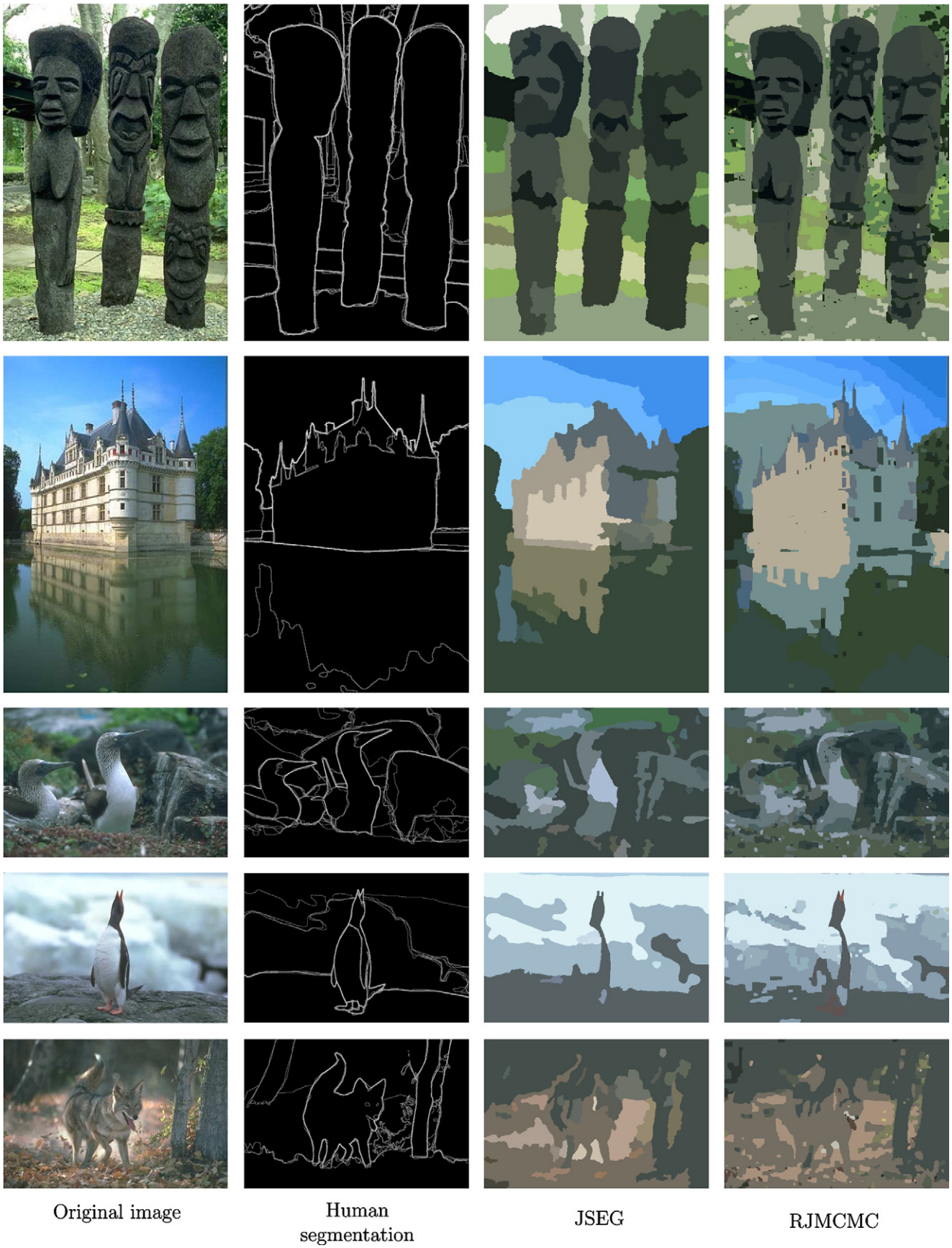


Fig. 7. Benchmark results on images from the Berkeley Segmentation Dataset.

Table 1
*F*-measure and CPU time comparison

| Method | *F*-measure | CPU time ($730 \times 500$ image) |
|---|---|---|
| Human segmentation | 0.79 | — |
| RJMCMC | 0.57 | 15 min |
| JSEG | 0.56 | 2 min |

cases. This is demonstrated in Fig. 6, where we plot precision-recall curves for $\beta = 2.5$, $\beta = 0.5$, and $\beta = 10.0$.

Independently of the input image, we start the algorithm with two classes ($\hat{L}^0 = 2$), each of them having equal weights ($\hat{p}_0^0 = \hat{p}_1^0 = 0.5$). The initial mean vectors were set to $[0.2, 0.2, 0.2]$ and $[0.7, 0.7, 0.7]$, and both covariance matrices were initialized as

$$\hat{\boldsymbol{\Sigma}}_0^0 = \hat{\boldsymbol{\Sigma}}_1^0 = \begin{pmatrix} 0.05 & 0.00001 & 0.00001 \\ 0.00001 & 0.05 & 0.00001 \\ 0.00001 & 0.00001 & 0.05 \end{pmatrix}$$

As an example, we show in Fig. 2 these initial Gaussians as well as the final estimates. In spite of the rough initialization, our algorithm finds the three meaningful classes and an accurate segmentation is obtained.

In subsequent figures, we will compare the proposed method to JSEG [36], which is a recent unsupervised color image segmentation algorithm. It consists of two independent steps:

(1) Colors in the image are quantized to several representative classes. The output is a class map where pixels are replaced by their corresponding color class labels.
(2) A region growing method is then used to segment the image based on the multi-scale *J-images*. A *J-image* is produced by applying a criterion to local windows in the class-map (see [36] for details on that).

JSEG is also region based, uses similar cues (color similarity and spatial proximity) than our method, and it is fully automatic. We have used the program provided by the authors [36] and kept its default settings throughout our test: automatic color quantization threshold and number of scales, the region merge threshold was also set to its default value (0.4). Note that JSEG is not model based, therefore there are no pixel classes. Regions are identified based on the underlying color properties of the input image. Although we also show the number of labels for JSEG in our test results, these numbers reflect the number of detected *regions*. In our method, however, the *same* label is assigned to spatially distant regions if they are modeled by the same Gaussian component. Segmentation results are displayed as a *cartoon image* where pixel values are replaced by their label's average color in order to help visual evaluation of the segmentation quality.

In Figs. 3–5, we show the results obtained on test images found at the Kodak Digital Image Offering website (http://www.kodak.com/digitalImaging/samples/imageIntro.shtml). Both methods gave accurate segmentations but fine details were better preserved by RJMCMC. In Fig. 7, we show a couple of results obtained on the Berkeley Segmentation Dataset [33], and in Fig. 6, we plot the corresponding precision-recall curves. Note that RJMCMC has a slightly higher *F-measure* (see Table 1) which ranks it over JSEG. However, it is fair to say that both method perform equally well but behave differently: while JSEG tends to smooth out fine details (hence it has a higher precision but lower recall value), RJMCMC prefers to keep fine details at the price of producing more edges (i.e. its recall values are higher at a lower precision value).

It has to be noted that RJMCMC requires a higher CPU time (see Table 1). However, the running time can be reduced by at least a factor of 2 when using a faster random number generator [37] (currently we are using the standard `drand48()` C function) and by replacing the `exp()` function calls with a faster approximation [38].

Finally, we found that the acceptance rate for the split or merge move was ≈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 [11].

## 6. Conclusion

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.

## Acknowledgements

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.

## References

[1] C.L. Huang, T.Y. Cheng, C.C. Chen, Color images segmentation using scale space filter and Markov random field, Pattern Recognition 25 (10) (1992) 1217–1229.
[2] D.K. Panjwani, G. Healey, Markov random field models for unsupervised segmentation of textured color images, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (10) (1995) 939–954.

[3] P.J. Green, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, Biometrika 82 (4) (1995) 711–731.
[4] N. Giordana, W. Pieczynski, Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (5) (1997) 465–475.
[5] L. Gupta, T. Sortrakul, A Gaussian-mixture-based image segmentation algorithm, Pattern Recognition 31 (3) (1998) 315–325.
[6] C.S. Won, H. Derin, Unsupervised segmentation of noisy and textured images using Markov random fields, Computer Graphics and Image Processing: Graphical Models and Image Processing 54 (4) (1992) 208–328.
[7] D.A. Langan, J.W. Modestino, J. Zhang, Cluster validation for unsupervised stochastic model-based image segmentation, IEEE Transactions on Image Processing 7 (2) (1998) 180–195.
[8] J. Liu, Y.H. Yang, Multiresolution color image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (7) (1994) 689–700.
[9] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1984) 721–741.
[10] S. Richardson, P.J. Green, On Bayesian analysis of mixtures with an unknown number of components, Journal of the Royal Statistical Society, series B 59 (4) (1997) 731–792.
[11] C.P. Robert, T. Rydén, 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) (2000) 57–75.
[12] S.A. Barker, P.J.W. Rayner, Unsupervised image segmentation using Markov random field models, Pattern Recognition 33 (4) (2000) 587–602.
[13] S.C. Zhu, Stochastic jump-diffusion process for computing medial axes in Markov random fields, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (11) (1999) 1158–1169.
[14] D. Kersten, P. Mamassian, A. Yuille, Object perception as Bayesian inference, Annual Review of Psychology 55 (2004) 271–304.
[15] D. Mumford, Pattern theory: a unifying perspective, in: D. Knill, W. Richards (Eds.), Perception as Bayesian Inference, Cambridge University Press, Cambridge, 1996, pp. 25–62.
[16] B. Chalmond, Modeling and Inverse Problems in Image Analysis, Springer, New York, 2003.
[17] G. Winkler, Image Analysis, Random Fields and Markov Chain Monte Carlo Methods, 2nd ed., Springer, New York, 2003.
[18] D. Mumford, The Bayesian rationale for energy functionals, in: B. Romeny (Ed.), Geometry-Driven Diffusion in Computer Vision, Kluwer Academic, Dordrecht, 1994, pp. 141–153.
[19] H. Permuter, J. Francos, I. Jermyn, A study of Gaussian mixture models of colour and texture features for image classification and segmentation, Pattern Recognition 39 (4) (2006) 695–706.
[20] G.J. Miao, M.A. Clements, Digital Signal Processing and Statistical Classification, Artech House, 2002, ISBN 1580531350.
[21] Z. Kato, T.C. Pong, J.C.M. Lee, Color image segmentation and parameter estimation in a Markovian framework, Pattern Recognition Letters 22 (3-4) (2001) 309–321.
[22] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society, series B 39 (1) (1977) 1–38.
[23] S. Lakshmanan, H. Derin, Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (8) (1989) 799–813.
[24] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their application, Biometrika 57 (1970) 97–109.
[25] P.J. Green, Trans-dimensional Markov chain Monte Carlo, in: P.J. Green, N.L. Hjort, S. Richardson (Eds.), Highly Structured Stochastic Systems, Oxford, 2003.
[26] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculations by fast computing machines, Journal of Chemical Physics, vol. 21, pp. 1087–1092.
[27] S.P. Brooks, P. Giudici, G.O. Roberts, Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions, Journal of the Royal Statistical Society, series B 65 (2003) 3–55.
[28] M. Stephens, Bayesian methods for mixtures of normal distributions, Ph.D. thesis, University of Oxford, 1997.
[29] M. Stephens, Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods, Annals of Statistics 28 (1) (2000) 40–74.
[30] D. Cremers, F. Tischhauser, J. Weickert, C. Schnorr, Diffusion snakes: Introducing statistical shape knowledge into the Mumford–Shah functional, International Journal of Computer Vision 50 (3) (2002) 295–313.
[31] K.V. Mardia, J.T. Kent, J.M. Bibby, Multivariate Analysis, Academic Press, Duluth, London, 1979.
[32] J. Besag, On the statistical analysis of dirty pictures, Journal of the Royal Statistical Society, series B 48 (3) (1986) 259–302.
[33] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings of International Conference on Computer Vision, vol. 2, IEEE, 2001, pp. 416–423.
[34] S.J. Sangwine, R.E.N. Horne (Eds.), The Colour Image Processing Handbook, Chapman & Hall, London, 1998.
[35] Z. Kato, Bayesian color image segmentation using reversible jump Markov chain Monte Carlo, Research Report 01/99-R055, ERCIM/CWI, Amsterdam, The Netherlands, also available as a CWI Research Report PNA-R9902, ISSN 1386-3711 (Jan. 1999). URL http://www.ercim.org/publication/technical_reports/055-%abstract.html.
[36] Y. Deng, B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (8) (2001) 800–810, URL http://vision.ece.ucsb.edu/segmentation/jseg/.
[37] M. Matsumoto, T. Nishimura, Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Transactions on Modeling and Computer Simulation 8 (1) (1998) 3–30.
[38] N.N. Schraudolph, A fast, compact approximation of the exponential function, Neural Computation 11 (1999) 853–862.