Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Bayesian color image segmentation using reversible jump Markov chain Monte Carlo

Z. Kato

Probability, Networks and Algorithms (PNA)

PNA-R9902 February 28, 1999

Report PNA-R9902
ISSN 1386-3711

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Bayesian Color Image Segmentation Using Reversible Jump Markov Chain Monte Carlo

Zoltan Kato
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

This paper deals with the problem of unsupervised image segmentation. Our goal is to propose a method which is able to segment a color image without any human intervention. The only input is the observed image, all other parameters are estimated during the segmentation process. Our method is model-based, we use a first order Markov random field (MRF) model (also known as the Potts model) where the singleton energies derive from a multivariate Gaussian distribution and second order potentials favor similar classes in neighboring pixels. The most difficult part is the estimation of the number of pixel classes or in other words, the estimation of the number of Gaussian mixture components. Reversible jump Markov chain Monte Carlo (MCMC) is used to solve this problem. These jumps enable the possible splitting and merging of classes. The 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) criteria. Experimental results are promising, we have obtained accurate results on a variety of real color images.

1991 Mathematics Subject Classification: 62F15, 62M40
1991 Computing Reviews Classification System: I.4.6, I.4.8, I.4.10
Keywords and Phrases: unsupervised image segmentation, color, parameter estimation, normal mixture identification, Markov random fields, Markov chain Monte Carlo, reversible jumps, simulated annealing
Note: This research was supported by an ERCIM postdoctoral fellowship

1. Introduction

Image segmentation is an important early vision task where pixels with similar features are grouped into homogeneous regions. Many high level processing tasks (surface description, object recognition, content based indexing and retrieval for example) is based on such a preprocessed image. Our approach consists in building a Bayesian pixel classification model and finding the most likely labeling of pixels (ie. segmentation). To do so, we need to define some probability measure on the set of all possible labelings. In real scenes, neighboring pixels usually have similar properties. In a probabilistic framework, such regularities are well expressed by MRF's. Another reason for dealing with MRF models is that such a modelization is the one which requires the less a priori information on the world model. As a matter of fact, the simplest statistical model for an image consists of the probabilities of pixel classes. The knowledge on the dependencies between nearby pixels is much more powerful, and imposes few constraints. In a way, it is difficult to conceive a more general model, 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.

Color image segmentation in a Markovian framework has been addressed by several researchers [1, 2, 3, 4, 5]. In these approaches, the color difference of neighboring pixels is used in the MRF model. The first question, when dealing with color images, is how to measure quantitatively color difference between any two arbitrary color [6]. Experimental evidence suggests that the RGB tristimulus color space may be considered as a Riemannian space [7]. Due to the complexity of determining color distance in such spaces, several simple formulas have been proposed. These formulas approximate the Riemannian space by a Euclidean color space yielding a perceptually uniform spacing of colors [6, 8]. Some examples of these formulas, that we use herein, are $L^*u^*v^*$ [7], LAB [7, 9] or LHS [7] color spaces. Another advantage of these color metrics is that luminance and chroma information is separated.

Due to the difficulty of estimating the number of pixel classes (or clusters), unsupervised algorithms often suppose that this parameter is *known a priori* [10, 11, 12, 13, 14]. 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 to this problem in the literature.

One of them is an exhaustive search of the combined parameter space[15, 16]. In [15], segmentation and parameter estimation 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. This process is repeated over different model-dimensions. The resulting estimates are then applied to a model fitting criteria to select the optimum number of classes. A similar approach is used in [16] where the EM algorithm [17] is used to estimate parameters. The optimum model order is then selected by fitting the function of increasing likelihood against increasing model dimension to a rising exponential model.

The second approach consists of a two step approximation technique [2, 3, 18, 19]: the first step is a coarse segmentation of the image into the most likely number of regions. The parameter values are estimated from the resulting segmentation and the final result is obtained via a supervised segmentation. A multi-resolution approach is presented in [2]. where a scale space filter (SSF) is used to determine significant peaks in the histogram. Then, the histogram clustering information is used to perform a coarse segmentation of the image. The final segmentation is then obtained trough an MRF model defined over a quad-tree structure. The MRF model is used in [2] to control a split and merge algorithm. A similar model is presented in [3] but they use a monogrid MRF model and the final segmentation is obtained trough simulated annealing. In [18, 19], the image is divided into windows. Then parameters are estimated and closely related windows are merged. The resulting segmentation is used to obtain final estimates for a supervised segmentation which is carried out via a relaxation algorithm. In [4], an unsupervised segmentation algorithm is proposed which uses MRF models for *color textures*. These models are defined in each color plane with interactions between different color planes. The segmentation algorithm is based on agglomerative hierarchical clustering but it also aims at maximizing the conditional pseudo-likelihood of the image given the regions and the MRF parameters. The algorithm consists of a region splitting phase followed by a conservative merging. Finally, a stepwise optimal merging process based on a global performance function is used to complete the segmentation.

Our approach consists of building a Bayesian color image model using a first order MRF (also known as the Potts model [20]) where the external field (ie. 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 [21], We are interested in the *posterior distribution* of the unknows given the observed image. The unknows comprise the hidden label field configuration, the Gaussian mixture parameters, MRF hyperparameter and the number of mixture components (or classes). Then a MCMC algorithm is used to sample from the whole posterior distribution in order to obtain a MAP estimate via simulated annealing [22]. However, classical MCMC methods are restricted to problems where the dimensionality of the parameter vector is fixed. Therefore, the estimation of the number of mixture components is not possible. Recently, a new method, called Reversible Jump MCMC (RJMCMC), has been proposed by Peter Green in [23]. This method makes it possible to construct reversible Markov chain samplers that jump between parameter subspaces of different dimensionality. RJMCMC allows 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.

We also refer to two papers [24, 25], that are closely related to our work. In [24], RJMCMC has been applied to univariate Gaussian mixture identification while in [25], it is applied to intensity based unsupervised image segmentation with a modelization similar to [21, 26].

The main contribution of this paper consists in formulating the RJMCMC algorithm in a more general context than in [25]. Observations are taken from a higher dimension space and classes are represented by multivariate Gaussian distributions. Although we present the model in the case of 3 dimension observations, it is straightforward to extend it to higher dimensions.

The remainder of the paper is organized as follows: In the next section, we define the segmentation model. Section 3 describes the RJMCMC algorithm which is then applied to our model in Section 4 and Section 5. We formulate a simulated annealing algorithm in Section 6 which aims at finding the MAP estimate of the unknowns. Parallelization methods are also discussed. Finally experimental results of a sequential implementation are presented.

2. Color Image Segmentation Model

Our approach consists of building a probabilistic pixel classification model using Markov random fields. The classes are represented by a multi-variate Gaussian distribution. A global smoothing constraint (parameterized by β) is introduced trough a first order MRF, also known as the Potts model [20]. Let us suppose that the observed image $\mathcal{F} = \{\vec{\boldsymbol{f}}_s | s \in \mathcal{S}, \forall i : 0 < \vec{\boldsymbol{f}}_s^{\,i} < 1\}$ consists of three spectral component values at each pixel s denoted by the vector $\vec{\boldsymbol{f}}_s$. The segmentation is done by assigning a label $\omega_s \in \Lambda = \{1, 2, \ldots, L\}$ to each site s. $\omega \in \Omega$ denotes a labeling (or segmentation), Ω is the set of all possible labelings.

Basically, we regard our image as a sample drawn from an unknown Gaussian mixture distribution. The goal of our analysis is inference about the number L of components, the component parameters $\Theta = \{\forall 1 \leq \lambda \leq L : \Theta_\lambda = (\vec{\boldsymbol{\mu}}_\lambda, \Sigma_\lambda)\}$, the components weights $p_\lambda (1 \leq \lambda \leq L)$, summing to 1, the clique potential (or inter-pixel interaction strength) β and the segmentation ω.

The joint distribution of the variables $L, p, \beta, \omega, \Theta, \mathcal{F}$ is given by:

$$P(L, p, \beta, \omega, \Theta, \mathcal{F}) = P(\omega, \mathcal{F} \mid \Theta, \beta, p, L) P(\Theta, \beta, p, L) \tag{2.1}$$

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:

$$P(\Theta, \beta, p, L) = P(\Theta) P(\beta) P(p) P(L) \tag{2.2}$$

Let us concentrate now on the posterior distribution of $(\mathcal{F}, \omega)$ which may be expressed:

$$P(\omega, \mathcal{F} \mid \Theta, \beta, p, L) = P(\mathcal{F} \mid \omega, \Theta, \beta, p, L) P(\omega \mid \Theta, \beta, p, L) \tag{2.3}$$

Before further proceeding, let us examine the above factorization. As we declared earlier, pixel classes are represented by a multivariate Gaussian distribution and the underlying MRF label process follows a Gibbs distribution defined over a first order neighborhood system (see Figure 1). Thus we can impose further conditional independences yielding:

$$\begin{aligned} P(\mathcal{F} \mid \omega, \Theta, \beta, p, L) &= P(\mathcal{F} \mid \omega, \Theta) = \\ &= \prod_{s \in \mathcal{S}} \left(\frac{1}{\sqrt{(2\pi)^3 \mid \Sigma_{\omega_s} \mid}} \exp\left(-\frac{1}{2} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s}) \Sigma_{\omega_s}^{-1} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s})^T \right) \right) \end{aligned} \tag{2.4}$$

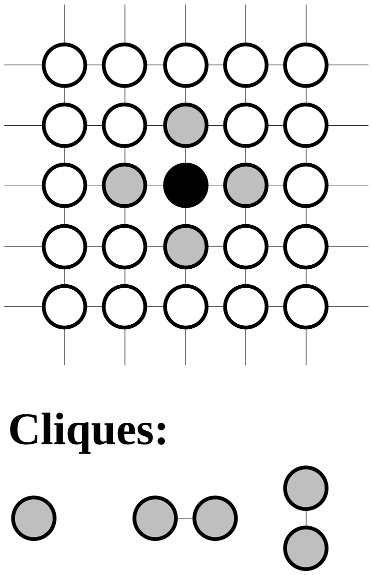


Figure 1: First order neighborhood system with cliques. Horizontal and vertical cliques are called doubletons. Cliques containing a single pixel are called singletons.

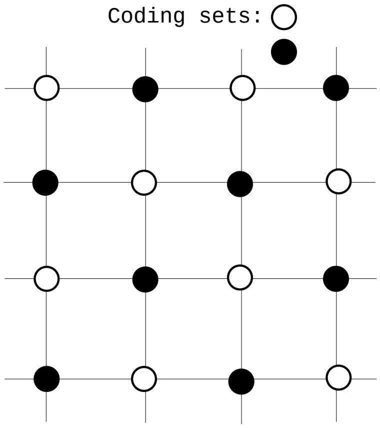


Figure 2: Coding sets in the case of a first order neighborhood system.

$$
\begin{aligned}
P(\omega \mid \Theta, \beta, p, L) &= P(\omega \mid \beta, p, L) \\
&= \frac{1}{Z(\beta, p, L)} \exp(-U(\omega \mid \beta, p, L)) \text{ , where} \qquad (2.5) \\
U(\omega \mid \beta, p, L) &= \sum_{s \in \mathcal{S}} -\log(p_{\omega_s}) + \beta \sum_{\{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r) \qquad (2.6)
\end{aligned}
$$

$U(\omega \mid \beta, p, L)$ 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 $\mathcal{C}$ denotes the set of cliques and $\{s, r\}$ is a doubleton (see Figure 1) containing the neighboring pixel sites s and r. We note that the whole posterior distribution can also be derived from a Gibbs distribution where the Gaussian distribution is taken into account in the energy of the external field:

$$
\begin{aligned}
&U(\mathcal{F} \mid \omega, \Theta) = -\log(P(\mathcal{F} \mid \omega, \Theta)) = \\
&= \sum_{s \in \mathcal{S}} \left(\ln\left(\sqrt{(2\pi)^3 \mid \Sigma_{\omega_s} \mid} \right) + \frac{1}{2} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s}) \Sigma_{\omega_s}^{-1} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s})^T \right) \qquad (2.7)
\end{aligned}
$$

Since the partition function $Z(\beta, p, L)$ is not tractable [27, 14], the comparison of the likelihood of two differing MRF realizations from Equation (2.5) is unfeasible. However, we can compare their Pseudo-Likelihood [28, 14]:

$$P(\omega \mid \beta, p, L) \approx \prod_{s \in \mathcal{S}} \frac{p_{\omega_s} \exp\left(-\beta \sum\limits_{\forall r:\{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r)\right)}{\sum\limits_{\lambda \in \Lambda} p_\lambda \exp\left(-\beta \sum\limits_{\forall r:\{s,r\} \in \mathcal{C}} \delta(\lambda, \omega_r)\right)} \tag{2.8}$$

Using Equation (2.1), Equation (2.2), Equation (2.4), Equation (2.8) and the fact that $P(\mathcal{F})$ is constant for a given image, we can now easily approximate the posterior density $P(L, p, \beta, \omega, \Theta \mid \mathcal{F}) = P(L, p, \beta, \omega, \Theta, \mathcal{F})/P(\mathcal{F})$:

$$\begin{aligned} P(L, p, \beta, \omega, \Theta \mid \mathcal{F}) &\approx P(\mathcal{F} \mid \omega, \Theta) P(\omega \mid \beta, p, L) P(\Theta) P(\beta) P(p) P(L) \\ \approx & \prod_{s \in \mathcal{S}} \left(\frac{1}{\sqrt{(2\pi)^3 \mid \Sigma_{\omega_s} \mid}} \exp\left(-\frac{1}{2}(\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s}) \Sigma_{\omega_s}^{-1} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s})^T\right)\right) \\ & \times \prod_{s \in \mathcal{S}} \frac{p_{\omega_s} \exp\left(-\beta \sum\limits_{\forall r:\{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r)\right)}{\sum\limits_{\lambda \in \Lambda} p_\lambda \exp\left(-\beta \sum\limits_{\forall r:\{s,r\} \in \mathcal{C}} \delta(\lambda, \omega_r)\right)} \\ & \times P(\beta) P(L) \prod_{\lambda \in \Lambda} P(\vec{\boldsymbol{\mu}}_\lambda) P(\Sigma_\lambda) P(p_\lambda) \end{aligned} \tag{2.9}$$

Concerning the priors, we will follow [24, 25] and chose uniform reference priors for L, $\vec{\boldsymbol{\mu}}_\lambda$, $\boldsymbol{\Sigma}_\lambda$, p_λ $(\lambda \in \Lambda)$.

3. Sampling from the Posterior Distribution

Herein, we shall construct an MCMC sampler which is used to sample from the posterior distribution of our segmentation model in Equation (2.9). Classical MCMC methods, however, can not be used cause of the changing dimensionality of the parameter space. Recently, MCMC methods for varying dimension problems have been discussed [29, 23]. A promising approach, called Reversible Jump MCMC (RJMCMC), is proposed in [23] and has been applied to the analysis of univariate normal mixtures in [24]. Let us denote the set of unknowns $\{L, p, \beta, \omega, \Theta\}$ by χ and let $\pi(\chi)$ be the target probability measure (the posterior distribution, in our context). A broadly used tool to sample from $\pi(\chi)$ is the Metropolis-Hastings method [30, 31]. When the current state is χ than a new state $\chi\prime$ is drawn form an essentially arbitrary joint distribution $q(\chi, \chi\prime)$. Then the new state is accepted with probability

$$\mathcal{A}(\chi, \chi\prime) = \min\left(1, \frac{\pi(\chi\prime) q(\chi, \chi\prime)}{\pi(\chi) q(\chi\prime, \chi)}\right) \tag{3.1}$$

When we have multiple parameter subspaces of different dimensionality, it is necessary to devise different *move types* between the subspaces [23]. These will be combined in a so called *hybrid sampler* [32] by random choice between available moves at each transition. Let us denote these move types by $m \in \mathcal{M} = \{1, 2, \ldots, M\}$ and let $q_m(\chi, \chi\prime)$ be the probability of proposing the move type m and state $\chi\prime$ when the current state is χ. We note that not all move types are necessarily available from all starting state, so for each χ, $q_m(\chi, \cdot)$ might be 0 for some m. Furthermore, $q_m(\chi, \cdot)$ is a sub-probability measure on m and $\chi\prime$. Thus

$$\sum_{m \in \mathcal{M}} q_m(\chi, \cdot) \leq 1. \tag{3.2}$$

The proposed state is then accepted with probability

$$\mathcal{A}_m(\chi, \chi\prime) = \min\left(1, \frac{\pi(\chi\prime) q_m(\chi, \chi\prime)}{\pi(\chi) q_m(\chi\prime, \chi)}\right) \tag{3.3}$$

For more details about a rigorous definition of the above ratio of measures, see [23]. Roughly speaking, the existence of such a measure is ensured by a *dimension matching* condition on $q_m(\chi\prime, \chi)$ that matches the degrees of freedom of joint variation of the current state χ and the proposal $\chi\prime$ as the dimension changes with L. We remark that for a move type which does not change the dimensionality of the parameter space, Equation (3.3) reduces to the usual Metropolis-Hastings acceptance probability (see Equation (3.1)). For dimension-changing moves, suppose that a move type m is proposed from χ to a state $\chi\prime$ in a higher dimensional space. Then the dimension matching can be implemented by drawing a vector of continuous random variables u and setting $\chi\prime$ to be some invertible deterministic function $\psi(\chi, u)$ of χ and u. The reverse move from $\chi\prime$ to χ can be accomplished by using the inverse transformation $\psi^{-1}(\chi\prime)$ so that the proposal is deterministic. The acceptance probability is then given by [23, 24]

$$\mathcal{A}_m(\chi, \chi\prime) = \min\left(1, \frac{\pi(\chi\prime) r_m(\chi\prime)}{\pi(\chi) r_m(\chi) q(u)} \left|\frac{\partial \chi\prime}{\partial(\chi, u)}\right|\right) \tag{3.4}$$

where $r_m(\chi)$ is the probability of choosing the move type m when in state χ and $q(u)$ is the density function of u. The last term is the absolute value of the Jacobian determinant of transformation ψ.

4. Hybrid Sampler

For our image segmentation model outlined in Section 2, we shall make use of five move types:

1. sampling the labels ω (ie. re-segment the image)
2. sampling Gaussian parameters $\Theta = \{(\vec{\boldsymbol{\mu}}_\lambda, \Sigma_\lambda) | \lambda \in \Lambda\}$;

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 in 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. Hereafter, we derive the sampling equations from Equation (2.9) for these move types. Concerning the fifth move type, the reversible jump mechanism is needed (see Section 3). We will discuss the equations of the split and combine move later, in Section 5.

4.1 Image Segmentation

This move type consists in a classical image segmentation with *known parameters.* Since the parameters L, p, β, Θ are set to their estimates $\widehat{L}$, $\widehat{p}$, $\widehat{\beta}$, $\widehat{\Theta}$, Equation (2.9) reduces to the following form:

$$\begin{aligned} P(L,p,\beta,\omega,\Theta \mid \mathcal{F}) &\approx P(\mathcal{F} \mid \omega, \widehat{\Theta}) P(\omega \mid \widehat{\beta}, \widehat{p}, \widehat{L}) \\ &\approx \prod_{s \in \mathcal{S}} \left(\frac{1}{\sqrt{(2\pi)^3 \mid \widehat{\Sigma}_{\omega_s} \mid}} \exp\left(-\frac{1}{2} (\vec{\boldsymbol{f}}_s - \widehat{\vec{\boldsymbol{\mu}}}_{\omega_s}) \widehat{\Sigma}_{\omega_s}^{-1} (\vec{\boldsymbol{f}}_s - \widehat{\vec{\boldsymbol{\mu}}}_{\omega_s})^T \right) \right) \\ &\quad \times \prod_{s \in \mathcal{S}} \widehat{p}_{\omega_s} \exp\left(-\widehat{\beta} \sum_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r) \right) \end{aligned} \tag{4.1}$$

Note that in this case there is no need to compute the partition function $Z(\beta, p, L) = Z(\widehat{\beta}, \widehat{p}, \widehat{L})$ as it is constant over Ω. Furthermore, the sub-chain can be sampled by a Gibbs sampler [21] since ω is discrete taking values from the finite set Λ.

4.2 Estimating Gaussian Parameters

The goal of this type of move is to estimate the mean vector and covariance matrix of the pixel classes. Setting variables L, p, β, ω to their estimates $\widehat{L}, \widehat{p}, \widehat{\beta}, \widehat{\omega}$, Equation (2.9) reduces to the following form:

$$\begin{aligned} P(L,p,\beta,\omega,\Theta \mid \mathcal{F}) &\approx P(\mathcal{F}, \widehat{\omega} \mid \Theta) P(\Theta) = \prod_{\lambda \in \Lambda} \prod_{s : \widehat{\omega}_s = \lambda} P(\vec{\boldsymbol{f}}_s \mid \vec{\boldsymbol{\mu}}_\lambda, \Sigma_\lambda) P(\vec{\boldsymbol{\mu}}_\lambda) P(\Sigma_\lambda) = \\ &= \prod_{\lambda \in \Lambda} \frac{1}{((2\pi)^3 \mid \Sigma_\lambda \mid)^{|\mathcal{S}_\lambda|/2}} \exp\left(-\frac{1}{2} \sum_{s : \widehat{\omega}_s = \lambda} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_\lambda) \Sigma_\lambda^{-1} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_\lambda)^T \right) \\ &\quad \times \prod_{\lambda \in \Lambda} P(\vec{\boldsymbol{\mu}}_\lambda) P(\Sigma_\lambda) P(p_\lambda) \end{aligned} \tag{4.2}$$

where $\mid \mathcal{S}_\lambda \mid$ is the number of sites labeled by λ.

4.3 Sampling Mixture Weights

This move type aims at estimating the mixture weights p. These weights are incorporated into the Gibbs distribution of the underlying label process ω as the external field strength. Furthermore, we require that the weights be normalized by imposing the following constraint:

$$\sum_{\lambda \in \Lambda} p_\lambda = 1 \tag{4.3}$$

In this way, it becomes possible to maintain a balance between the external and internal field strength. This makes it also possible to set the hyperparameter β *a priori* [25], an important point outlined in the next section. Using the above condition and setting variables L, β, ω, Θ to their estimates $\widehat{L}, \widehat{\beta}, \widehat{\omega}, \widehat{\Theta}$, we get:

$$\begin{aligned}
P(L, p, \beta, \omega, \Theta \mid \mathcal{F}) &\approx P(\widehat{\omega} \mid \widehat{\beta}, p, \widehat{L}) P(p) = \\
&= \prod_{\lambda \in \Lambda} P(p_\lambda) \left(\frac{p_\lambda}{\sum\limits_{\lambda \in \Lambda} p_\lambda} \right)^{|\mathcal{S}_\lambda|} \prod_{s \in \mathcal{S}} \frac{\exp\left(-\widehat{\beta} \sum\limits_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r) \right) \sum\limits_{\lambda \in \Lambda} p_\lambda}{\sum\limits_{\lambda \in \Lambda} p_\lambda \exp\left(-\widehat{\beta} \sum\limits_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\lambda, \widehat{\omega}_r) \right)} \\
&= \prod_{\lambda \in \Lambda} P(p_\lambda) p_\lambda^{|\mathcal{S}_\lambda|} \prod_{s \in \mathcal{S}} \frac{\exp\left(-\widehat{\beta} \sum\limits_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r) \right)}{\sum\limits_{\lambda \in \Lambda} p_\lambda \exp\left(-\widehat{\beta} \sum\limits_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\lambda, \widehat{\omega}_r) \right)}
\end{aligned} \tag{4.4}$$

4.4 Sampling Hyperparameter β

In this case, L, p, ω, Θ are set to their estimates $\widehat{L}, \widehat{p}, \widehat{\omega}, \widehat{\Theta}$. Equation (2.9) reduces to the following form:

$$\begin{aligned}
P(L, p, \beta, \omega, \Theta \mid \mathcal{F}) \;\approx\; & P(\widehat{\omega} \mid \beta, \widehat{p}, \widehat{L}) P(\beta) = \\
= & \prod_{s \in \mathcal{S}} \frac{\exp\left(-\widehat{\beta} \sum\limits_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r) \right)}{\sum\limits_{\lambda \in \Lambda} p_\lambda \exp\left(-\widehat{\beta} \sum\limits_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\lambda, \widehat{\omega}_r) \right)} P(\beta)
\end{aligned} \tag{4.5}$$

Unfortunately, due to the consequences of approximating the likelihood by the pseudo-likelihood, the posterior density of β will not be proper under particular label configurations [25]. Here, we will follow [25] and fix β *a priori*.

5. Estimating Number of Classes

Herein, we shall discuss the split and combine move. Recall, that 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 Equation (2.9) is sampled. Only β can be set to his estimate $\widehat{\beta}$ yielding the following distribution:

$$\begin{aligned}
P(L, p, \beta, \omega, \Theta \mid \mathcal{F}) &\approx P(\mathcal{F} \mid \omega, \Theta) P(\omega \mid \widehat{\beta}, p, L) P(\Theta) P(p) P(L) \\
&\approx \prod_{s \in \mathcal{S}} \left(\frac{1}{\sqrt{(2\pi)^3 \mid \Sigma_{\omega_s} \mid}} \exp\left(-\frac{1}{2} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s}) \Sigma_{\omega_s}^{-1} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s})^T \right) \right) \\
&\times \prod_{s \in \mathcal{S}} \frac{p_{\omega_s} \exp\left(-\widehat{\beta} \sum\limits_{\forall r: \{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r) \right)}{\sum\limits_{\lambda \in \Lambda} p_\lambda \exp\left(-\widehat{\beta} \sum\limits_{\forall r: \{s,r\} \in \mathcal{C}} \delta(\lambda, \omega_r) \right)} \\
&\times P(L) \prod_{\lambda \in \Lambda} P(\vec{\boldsymbol{\mu}}_\lambda) P(\Sigma_\lambda) P(p_\lambda) \qquad (5.1)
\end{aligned}$$

Since the dimensionality of the parameter space is altered, the reversible jump technique, outlined in Section 3, is needed for sampling from the posterior distribution.

First, we randomly choose between proposing to increment (*split*) or decrement (*merge*) the number of classes L, with probability $P_{split}(L)$ and $P_{merge}(L) = 1 - P_{split}(L)$ respectively. If we denote the maximum value allowed for L by L_{max} and the minimum value by L_{min} than we set:

$$P_{split}(L) = \begin{cases} 0 & \text{if } L = L_{max} \\ 1 & \text{if } L = L_{min} \\ 0.5 & \text{otherwise} \end{cases} \qquad (5.2)$$

$$P_{merge}(L) = \begin{cases} 0 & \text{if } L = L_{min} \\ 1 & \text{if } L = L_{max} \\ 0.5 & \text{otherwise} \end{cases} \qquad (5.3)$$

5.1 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 splitted into λ_1 and λ_2. In doing so, a new set of parameters need to be generated. Recall from Section 3 that we have to define a deterministic function $\psi(\chi, u)$ in order to satisfy the *dimension matching* constraint on the proposal probability $q_m(\chi\prime, \chi)$. In our context, altering L changes the dimensionality of the variables Θ and p. Thus we shall define ψ as a function of these Gaussian mixture parameters:

$$(\Theta^+, p^+) = \psi(\Theta, p, u) \qquad (5.4)$$

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^+).

Generating New Parameters Herein, we define ψ using a similar approach to the one described in [24]: The new parameters are assigned by matching the 0^{th}, 1^{th}, 2^{th} moments of the component being splitted to those of a combination of the two new components.

$$\begin{aligned} p_\lambda &= p^+_{\lambda_1} + p^+_{\lambda_2} && (5.5)\\ p_\lambda \vec{\boldsymbol{\mu}}_\lambda &= p^+_{\lambda_1}\vec{\boldsymbol{\mu}}^+_{\lambda_1} + p^+_{\lambda_2}\vec{\boldsymbol{\mu}}^+_{\lambda_2} && (5.6)\\ p_\lambda(\vec{\boldsymbol{\mu}}_\lambda \vec{\boldsymbol{\mu}}_\lambda^T + \boldsymbol{\Sigma}_\lambda) &= p^+_{\lambda_1}(\vec{\boldsymbol{\mu}}^+_{\lambda_1}\vec{\boldsymbol{\mu}}^{+T}_{\lambda_1} + \boldsymbol{\Sigma}^+_{\lambda_1}) + p^+_{\lambda_2}(\vec{\boldsymbol{\mu}}^+_{\lambda_2}\vec{\boldsymbol{\mu}}^{+T}_{\lambda_2} + \boldsymbol{\Sigma}^+_{\lambda_2}) && (5.7) \end{aligned}$$

We remark that the above definition of ψ leaves unchanged the parameters of the classes other than the class being splitted. Thus Equation (5.4) could be written as:

$$(\Theta^+_\lambda, p^+_\lambda) = \psi(\Theta_{\lambda_1}, \Theta_{\lambda_2}, p_{\lambda_1}, p_{\lambda_2}, u) \qquad (5.8)$$

There are 10 degrees of freedom in splitting λ conforming to Equations (5.5)–(5.7) since covariance matrices are symmetric. Therefore we need to generate the following set of random variables:

$$u1, \quad \vec{\boldsymbol{u2}} = [u\mathit{2}_1, u\mathit{2}_2, u\mathit{2}_3], \quad \boldsymbol{u3} = \begin{bmatrix} u\mathit{3}_{1,1} & u\mathit{3}_{1,2} & u\mathit{3}_{1,3} \\ u\mathit{3}_{1,2} & u\mathit{3}_{2,2} & u\mathit{3}_{2,3} \\ u\mathit{3}_{1,3} & u\mathit{3}_{2,3} & u\mathit{3}_{3,3} \end{bmatrix} \qquad (5.9)$$

We can now compute the new parameter values:

$$\begin{aligned} p^+_{\lambda_1} &= p_\lambda u1 && (5.10)\\ p_{\lambda_2} &= p_\lambda(1 - u1) && (5.11)\\ \mu^+_{\lambda_1,i} &= \mu_{\lambda,i} + u\mathit{2}_i\sqrt{\Sigma_{\lambda,i,i}\frac{1 - u\mathit{1}}{u\mathit{1}}} && (5.12)\\ \mu^+_{\lambda_2,i} &= \mu_{\lambda,i} - u\mathit{2}_i\sqrt{\Sigma_{\lambda,i,i}\frac{u\mathit{1}}{1 - u\mathit{1}}} && (5.13)\\ \Sigma^+_{\lambda_1,i,j} &= \begin{cases} u\mathit{3}_{i,i}\left(1 - u\mathit{2}_i{}^2\right)\Sigma_{\lambda,i,i}\dfrac{1}{u\mathit{1}} & \text{if } i = j \\ u\mathit{3}_{i,j}\Sigma_{\lambda,i,j}\sqrt{\left(1 - u\mathit{2}_i{}^2\right)\left(1 - u\mathit{2}_j{}^2\right)u\mathit{3}_{i,i}u\mathit{3}_{j,j}} & \text{if } i \neq j \end{cases} && (5.14) \end{aligned}$$

$$\Sigma^+_{\lambda_2,i,j} = \qquad (5.15)$$

$$\begin{cases} \left(1 - u\mathit{3}_{i,i}\right)\left(1 - u\mathit{2}_i{}^2\right)\Sigma_{\lambda,i,i}\dfrac{1}{u\mathit{1}} & \text{if } i = j \\ \left(1 - u\mathit{3}_{i,j}\right)\Sigma_{\lambda,i,j}\sqrt{\left(1 - u\mathit{2}_i{}^2\right)\left(1 - u\mathit{2}_j{}^2\right)\left(1 - u\mathit{3}_{i,i}\right)\left(1 - u\mathit{3}_{j,j}\right)} & \text{if } i \neq j \end{cases}$$

The random variables are chosen from the interval $(0, 1]$. In order to favor splitting the class into roughly equal portions, beta distributions are used:

$$u1 \sim \mathrm{beta}(1.1, 1.1), \quad u2_i \sim \mathrm{beta}(1.1, 1.1), \quad u3_{i,j} \sim \mathrm{beta}(1.1, 1.1) \tag{5.16}$$

Reallocation of the Labels Once the new parameters for λ_1 and λ_2 have been computed, we have to propose the reallocation of those sites $s \in \mathcal{S}_\lambda$ where $\widehat{\omega}_s = \lambda$. 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 in Equation (4.1) with $\widehat{\Theta} = \Theta^+$, $\widehat{p} = p^+$ and $\widehat{L} = L + 1$. At the moment of splitting, the neighborhood configuration at a given site $s \in \mathcal{S}_\lambda$ 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 $\mathcal{S}_\lambda$ then we can sample the posterior distribution in Equation (4.1) 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 [28] algorithm to obtain a suboptimal segmentation of $\mathcal{S}_\lambda$. The resulting label map can be used to draw a sample from Equation (4.1) using a one step Gibbs sampler. The obtained ω^+ has a relatively high posterior probability since the tentative labeling was close the optimal labeling.

5.2 Merging Two Classes

First of all, we have to decide which pair of classes has to be merged. A pair (λ_1, λ_2) is chosen with probability relative to their distance:

$$P^{merge}_{select}(\lambda_1, \lambda_2) = \frac{d(\lambda_1, \lambda_2)}{\displaystyle\sum_{\lambda \in \Lambda} \sum_{\kappa \in \Lambda} d(\lambda, \kappa)} \tag{5.17}$$

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

$$d(\lambda_1, \lambda_2) = (\vec{\boldsymbol{\mu}}_{\lambda_1} - \vec{\boldsymbol{\mu}}_{\lambda_2})\boldsymbol{\Sigma}^{-1}_{\lambda_1}(\vec{\boldsymbol{\mu}}_{\lambda_1} - \vec{\boldsymbol{\mu}}_{\lambda_2}) + (\vec{\boldsymbol{\mu}}_{\lambda_2} - \vec{\boldsymbol{\mu}}_{\lambda_1})\boldsymbol{\Sigma}^{-1}_{\lambda_2}(\vec{\boldsymbol{\mu}}_{\lambda_2} - \vec{\boldsymbol{\mu}}_{\lambda_1}) \tag{5.18}$$

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 Equations (5.5)–(5.7). The reallocation is simply done by setting the label at sites $s \in \mathcal{S}_{\{\lambda_1,\lambda_2\}}$ to the new label λ. The random variables u are obtained by back-substitution into Equations (5.10)–(5.15).

5.3 Acceptance Probability

The acceptance probabilities for the split or merge moves can be calculated from Equation (3.4). Let us first consider the probability 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.

$$\mathcal{A}_{split}(L, \widehat{p}, \widehat{\beta}, \widehat{\omega}, \widehat{\Theta}; L+1, p^{+}, \widehat{\beta}, \omega^{+}, \Theta^{+}) = \min(1, A) \tag{5.19}$$

where

$$\begin{aligned} A = & \frac{P(L+1, p^{+}, \widehat{\beta}, \omega^{+}, \Theta^{+} \mid \mathcal{F})}{P(L, \widehat{p}, \widehat{\beta}, \widehat{\omega}, \widehat{\Theta} \mid \mathcal{F})} \frac{P_{merge}(L+1) P_{select}^{merge}(\lambda_1, \lambda_2)}{P_{split}(L) P_{select}^{split}(\lambda) P_{realloc}} \\ \times & \frac{1}{P_{beta(1.1,1.1)}(u1) \prod_{i=1}^{3} \left(P_{beta(1.1,1.1)}(u2_i) \prod_{j=i}^{3} P_{beta(1.1,1.1)}(u3_{i,j}) \right)} \left| \frac{\partial \psi}{\partial(\Theta_\lambda, p_\lambda, u)} \right| \end{aligned} \tag{5.20}$$

$P_{realloc}$ denotes the probability of reallocating pixels labeled by λ into regions labeled by λ_1 and λ_2. It can be derived from Equation (4.1) by restricting the set of labels Λ^{+} to the subset $\{\lambda_1, \lambda_2\}$ and taking into account only those sites s for which $\omega_s^{+} \in \{\lambda_1, \lambda_2\}$:

$$\begin{aligned} & P_{realloc} \approx \\ & \prod_{\forall s : \omega_s^{+} \in \{\lambda_1, \lambda_2\}} \left(\frac{1}{\sqrt{(2\pi)^3 \mid \mathbf{\Sigma}_{\omega_s^{+}}^{+} \mid}} \exp\left(-\frac{1}{2} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s^{+}}^{+}) \mathbf{\Sigma}_{\omega_s^{+}}^{+^{-1}} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s^{+}}^{+})^T \right) \right) \\ \times & \prod_{\forall s : \omega_s^{+} \in \{\lambda_1, \lambda_2\}} p_{\omega_s^{+}}^{+} \exp\left(-\widehat{\beta} \sum_{\forall r : \{s, r\} \in \mathcal{C}} \delta(\omega_s^{+}, \omega_r^{+}) \right) \end{aligned} \tag{5.21}$$

The correspondence between Equation (3.4) and Equation (5.20) is fairly straightforward:

$$\pi(\chi\prime) = P(L+1, p^{+}, \widehat{\beta}, \omega^{+}, \Theta^{+} \mid \mathcal{F}) \tag{5.22}$$

$$\pi(\chi) = P(L, \widehat{p}, \widehat{\beta}, \widehat{\omega}, \widehat{\Theta} \mid \mathcal{F}) \tag{5.23}$$

$$r_{merge}(\chi\prime) = P_{merge}(L+1) P_{select}^{merge}(\lambda_1, \lambda_2) \tag{5.24}$$

$$r_{split}(\chi) = P_{split}(L) P_{select}^{split}(\lambda) P_{realloc} \tag{5.25}$$

$$q(u) = P_{beta(1.1,1.1)}(u1) \prod_{i=1}^{3} \left(P_{beta(1.1,1.1)}(u2_i) \prod_{j=i}^{3} P_{beta(1.1,1.1)}(u3_{i,j}) \right) \tag{5.26}$$

The last factor is the Jacobian determinant of the transformation ψ:

$$\begin{aligned} & \left| \frac{\partial \psi}{\partial(\Theta_\lambda, p_\lambda, u)} \right| = \left| \frac{\partial(\Theta_{\lambda_1}^{+}, p_{\lambda_1}^{+}, \Theta_{\lambda_2}^{+}, p_{\lambda_2}^{+})}{\partial(\Theta_\lambda, p_\lambda, u)} \right| = \\ = & -w \prod_{i=1}^{3} \left(\frac{\Sigma_{i,i}^{2}}{u1 \ (u1 - 1)} \left(1 - u2_i^{\ 2}\right) (1 - u3_{i,i}) \, u3_{i,i} \prod_{j=i}^{3} \frac{\Sigma_{i,j}}{u1 \ (u1 - 1)} \right) \end{aligned} \tag{5.27}$$

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

$$\mathcal{A}_{merge}(L, \widehat{p}, \widehat{\beta}, \widehat{\omega}, \widehat{\Theta}; L-1, p^{-}, \widehat{\beta}, \omega^{-}, \Theta^{-}) = \min\left(1, \frac{1}{A}\right) \tag{5.28}$$

6. Optimization According to the MAP Criteria

In this section, we shall build a MAP estimator that provide us with an optimal segmentation $\widehat{\omega}$ and model parameters $\widehat{L}, \widehat{p}, \widehat{\beta}, \widehat{\Theta}$. The estimation is done via a stochastic relaxation algorithm. The MAP estimator of the unknowns is given by:

$$(\widehat{\omega}, \widehat{L}, \widehat{p}, \widehat{\beta}, \widehat{\Theta})^{(MAP)} = \arg \max_{L,p,\beta,\omega,\Theta} P(L, p, \beta, \omega, \Theta \mid \mathcal{F}) \tag{6.1}$$

with the following constraints:

$$\omega \in \Omega, \tag{6.2}$$

$$L_{min} \leq L \leq L_{max}, \tag{6.3}$$

$$\sum_{\lambda \in \Lambda} p_\lambda = 1, \tag{6.4}$$

$$\forall \lambda \in \Lambda : 0 \leq \mu_{\lambda,i} \leq 1, \tag{6.5}$$

$$\forall \lambda \in \Lambda : 0 \leq \Sigma_{\lambda,i,i} \leq 1, -1 \leq \Sigma_{\lambda,i,j} \leq 1 \tag{6.6}$$

Equation (6.1) is a combinatorial optimization problem which requires special algorithms such as simulated annealing [22]. In our case, simulated annealing can be formulated in the following way:

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 τ_0.*

② *A sample $(\widehat{\omega}^k, \widehat{L}^k, \widehat{p}^k, \widehat{\beta}^k, \widehat{\Theta}^k)$ is drawn from a slightly modified posterior distribution using the* hybrid sampler *defined in Section 4. The modification consists in including the* temperature *parameter τ_k:*

$$\prod_{s \in \mathcal{S}} \left(\frac{1}{((2\pi)^3 \mid \Sigma_{\omega_s} \mid)^{1/2\tau_k}} \exp\left(-\frac{1}{2\tau_k} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s}) \Sigma_{\omega_s}^{-1} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s})^T \right) \right)$$

$$\times \prod_{s \in \mathcal{S}} \frac{\exp\left(\dfrac{log(p_{\omega_s})}{\tau_k} - \dfrac{\beta}{\tau_k} \displaystyle\sum_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r) \right)}{\displaystyle\sum_{\lambda \in \Lambda} \exp\left(\dfrac{log(p_\lambda)}{\tau_k} - \dfrac{\beta}{\tau_k} \displaystyle\sum_{\forall r : \{s,r\} \in \mathcal{C}} \delta(\lambda, \omega_r) \right)}$$

The modification of the sub-chains' equations discussed in Section 4 and Section 5 is straightforward from the above equation. The sub-chains are then sampled via the following move-types:

1. *$\widehat{\omega}^k$ is drawn from the distribution in Equation (4.1),*
2. *$\widehat{\Theta}^k$ is obtained by sampling Equation (4.2),*
3. *$\widehat{p}^k$ is drawn from Equation (4.4) and*
4. *sampling the MRF hyperparameter $\widehat{\beta}^k$ from Equation (4.5);*
5. *$\widehat{L}^k$ estimated using the reversible jump technique presented in Section 5.*

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

There are different methods to implement the above algorithm in parallel. The most natural way is to explore data parallelism which permits the update of conditionally independent variables at the same time:

- Moves (2) and (3) can be performed in parallel since $(\mathbf{\Sigma}_\lambda, \vec{\boldsymbol{\mu}}_\lambda)$ and p_λ are conditionally independent given all other variables.
- For each λ: $\mathbf{\Sigma}_\lambda$, $\vec{\boldsymbol{\mu}}_\lambda$ and p_λ can be sampled in parallel.
- ω can also be sampled in parallel using a partially synchronous scheme: update only conditionally independent pixels at the same time. These pixels can be grouped into a so called *coding set* [33, 28, 21, 34] (see Figure 2).

7. Experimental Results

The proposed algorithm has been tested on a variety of real color images We have implemented the plain sequential version in C but the code has not been optimized. Herein, we present a few examples of these results. First, the original images were converted from RGB to LUV, LAB or LHS using the equations from [7]. The dynamic range of all color components was normalized such that they take their values from $(0, 1)$.

Independently of the input image, we began the algorithm with two classes, each of them having the following weight, mean vector and covariance matrix:

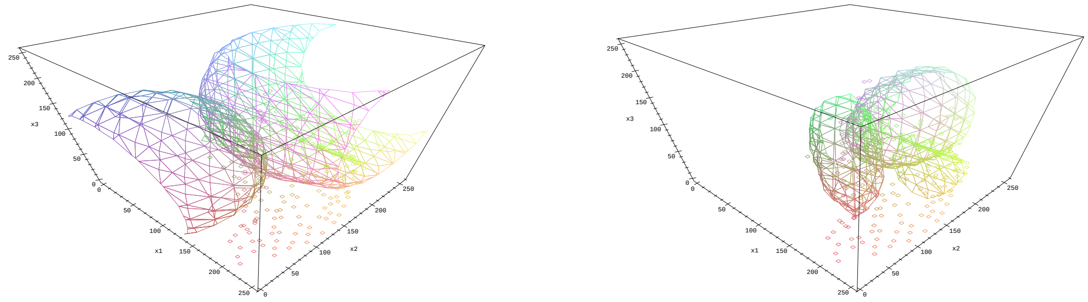
$$\widehat{L}^0 = 2 \tag{7.1}$$

$$\widehat{p}_0^0 = \widehat{p}_1^0 = 0.5 \tag{7.2}$$

$$\widehat{\vec{\boldsymbol{\mu}}_0}^0 = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \end{pmatrix} \tag{7.3}$$

$$\widehat{\vec{\boldsymbol{\mu}}_1}^0 = \begin{pmatrix} 0.7 \\ 0.7 \\ 0.7 \end{pmatrix} \tag{7.4}$$

$$\widehat{\mathbf{\Sigma}_0}^0 = \widehat{\mathbf{\Sigma}_1}^0 = \begin{pmatrix} 0.05 & 0.00001 & 0.00001 \\ 0.00001 & 0.05 & 0.00001 \\ 0.00001 & 0.00001 & 0.05 \end{pmatrix} \tag{7.5}$$



Initial Gaussians

Final Gaussians

Figure 3: Histogram, and the initial and final class parameters of image *rose41*

The hyperparameter β was fixed a priori. Therefore, move type (4) was not executed in Step ②. In our experiments, we found that $\beta = 2.5$ gives good results on all of the tested images. As usual, a logarithmic annealing schedule was chosen so that the algorithm would converge after a reasonable number of iterations. ($\mathcal{K}$ was set to 200 in our experiments). The schedule is given by:

$$T_{k+1} = 0.98T_k \tag{7.6}$$

with an initial temperature τ_0 set to 6.0. The final parameters and segmentation is then obtained trough the *RJMCMC Segmentation* algorithm. In proposing a *split* (see Section 5.1), an ICM algorithm [28] has been used to reallocate the labels by executing maximum 10 iterations. The label field ω was sampled by a Gibbs sampler [21] and the other parameters by a Metropolis-Hastings sampler [30, 31]. The algorithm was stoped after 200 iterations ($\tau_{200} \approx 0.1$). We note that small temperatures may cause an overflow in the computation of the posterior probability even when using double precision numbers and working with logarithms. This is one of the reasons why we only execute 200 iterations.

Figure 3 shows the histogram of the image in Figure 4 with the initial and final Gaussian parameters. We can see that the clustering of the histogram is a rather difficult task, classes are quite close to each other. The segmentation (see Figure 4) is accurate and the number of classes has been correctly diagnosed.

We have obtained similar results in Figure 5, Figure 6 and Figure 7. The latter result needs some explanation since the yellow region in the center of the postcard has not been correctly segmented. This is due to the rather noisy nature of this region and the relative small distance between the class parameters of green and yellow pixels. Where green and yellow pixels are mixed, the homogeneity constraint forces the formation of homogeneous regions and thus the misclassification of the yellow pixels as green. A more difficult scene is segmented in Figure 8, the result is reasonably accurate.

In the next set of images (Figure 9, Figure 10 and Figure 11), we can see that different color spaces may lead to slightly different results. Figure 9 shows the results obtained in the LUV and LHS color space. In Figure 11, results in the LUV and LAB spaces are presented. Finally, Figure 10 shows the result using LAB and LHS color metric.

In Figure 12, Figure 13, Figure 14 and Figure 15, some background-foreground style segmentation is shown. Although the foreground is quite accurately detected, we get more classes in the background that we would expect. The reason is that we can observ slight variation in the background color so that this variation is spatially characterized. For example, we can see in Figure 12 that the background is lighter in the middle of the image than at the corners. Therefore, when the background class is splitted into two, the reallocation process is able to form two homogeneous classes and the split is accepted. This problem could be solved by executing more iterations or by introducing an informative prior on the class parameters which favors well separated class mean values.

8. Conclusion

In this paper, we have proposed an unsupervised color image segmentation algorithm. We have established a Bayesian segmentation model using MRF modelization 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 an RJMCMC sampler. A single parameter is defined a priori which defines the interaction strength of neighboring pixels. The final estimates, satisfying the MAP criteria, are obtained trough simulated annealing. Experimental results show that an accurate segmentation can be obtained on a variety of real images.

Acknowledgment

The test images, used for evaluating the performance of the proposed algorithm, can be found at the following WWW sites: Kodak Digital Image Offering WWW site at
`http://www.kodak.com/digitalImaging/samples/imageIntro.shtml`,
INRIA-Syntim test images at
`http://www-syntim.inria.fr/syntim/analyse/images-eng.html`,
and the IEN database at
`http://www.ien.it/iengf/is/vislib.html`.
We also would like to thank to Simon Barker for providing us with the source code of his segmentation algorithm presented in [25].

References

1. M. J. Daily, “Color image segmentation using Markov random fields,” in *Proc. DARPA Image Understanding*, 1989.
2. J. Liu and Y. H. Yang, “Multiresolution Color Image Segmentation,” *IEEE-PAMI*, vol. 16, no. 7, pp. 689–700, July 1994.
3. C. L. Huang, T. Y. Cheng, and C. C. Chen, “Color images segmentation using scale space filter and Markov random field,” *Pattern Recognition*, vol. 25, no. 10, pp. 1217–1229, 1992.
4. D. K. Panjwani and G. Healey, “Markov Random Field Models for Unsupervised Segmentation of Textured Color Images,” *IEEE - PAMI*, vol. 17, no. 10, pp. 939–954, Oct. 1995.
5. Z. Kato, T. C. Pong, and J. C. M. Lee, “Motion compensated color video classification using Markov random fields,” in *Proc. ACCV*, R. Chin and T. C. Pong, Eds., Hong Kong, China, Jan. 1998, vol. I, pp. 738–745, Springer.
6. G. Healey and T. O. Binford, “A color metric for computer vision,” in *Proc. DARPA Image Understanding*, 1988, pp. 854–861.
7. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
8. P. Golland and A. M. Bruckstein, “Why R.G.B.? Or how to design color displays for Martians,” *Graphical Models and Image Processing*, vol. 58, no. 5, pp. 405–412, Sept. 1996.
9. C. Connolly and T. Fliess, “A study of efficiency and accuracy in the transformation from RGB to CIELAB color space,” *IEEE Trans. on Image Processing*, vol. 6, no. 7, pp. 1046–1048, July 1997.
10. P. Masson and W. Pieczynski, “SEM Algorithm and Unsupervised Statistical

Segmentation of Satellite Images," *IEEE Geoscience and Remote Sensing*, vol. 31, no. 3, pp. 618–633, May 1993.

11. N. Giordana and W. Pieczynski, "Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 465–475, May 1997.

12. E. Littmann and H. Ritter, "Adaptive color segmentation–a comparison of neural and statistical methods," *IEEE Trans. on Neural Networks*, vol. 8, no. 1, pp. 175–185, Jan. 1997.

13. L. Gupta and T. Sortrakul, "A Gaussian-mixture-based image segmentation algorithm," *Pattern Recognition*, vol. 31, no. 3, pp. 315–325, 1998.

14. 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*, vol. 11, no. 8, pp. 799–813, Aug. 1989.

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*, vol. 54, no. 4, pp. 308–328, July 1992.

16. D. A. Langan, J. W. Modestino, and J. Zhang, "Cluster validation for unsupervised stochastic model-based image segmentation," *IEEE Trans. on Image Processing*, vol. 7, no. 2, pp. 180–195, Feb. 1998.

17. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Jl. Roy. Statist. Soc., ser. B*, vol. vol. 39, no. 1, pp. 1–38, 1977.

18. H. H. Nguyen and P. Cohen, "Gibbs random fields, fuzzy clustering, and the unsupervised segmentation of textured images," *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, vol. 55, no. 1, pp. 1–19, Jan. 1993.

19. F. S. Cohen and Z. Fan, "Maximum likelihood unsupervised textured image segmentation," *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, vol. 54, no. 3, pp. 239–251, May 1992.

20. R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*, Academic Press, 1990.

21. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

22. P.J. Van Laarhoven and E.H. Aarts, "Simulated annealing : Theory and applications," *Reidel Pub., Dordrecht, Holland*, 1987.

23. P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–731, 1995.

24. S. Richardson and P. J. Green, "On Bayesian analysis of mixtures with an unknown number of components," *Jl. Roy. Statist. Soc., ser. B*, vol. 59, no. 4, pp. 731–792, 1997.

25. S. A. Barker and P. J. W. Rayner, "Unsupervised image segmentation using Markov random field models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition.* 1997, pp. 165–178, Springer.

26. Z. Kato, J. Zerubia, and M. Berthod, "Bayesian image classification using markov random fields," in *Maximum Entropy and Bayesian Methods*, A. Mohammad-Djafari G. Demoments, Ed. 1993, pp. 375–382, Kluwer Academic Publisher.

27. D. Geman, "Bayesian image analysis by adaptive annealing," in *Proc. IGARSS'85*, Amherst, USA, Oct. 1985, pp. 269–277.

28. J. Besag, "On the statistical analysis of dirty pictures," *Jl. Roy. Statist. Soc., ser. B*, 1986.

29. U. Grenander and M. Miller, "Representations of knowledge in complex systems," *Jl. Roy. Statist. Soc., ser. B*, vol. 56, pp. 549–603, 1994.

30. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. of Chem. Physics, Vol. 21, pp 1087-1092*, 1953.

31. W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their application," *Biometrika*, vol. 57, pp. 97–109, 1970.

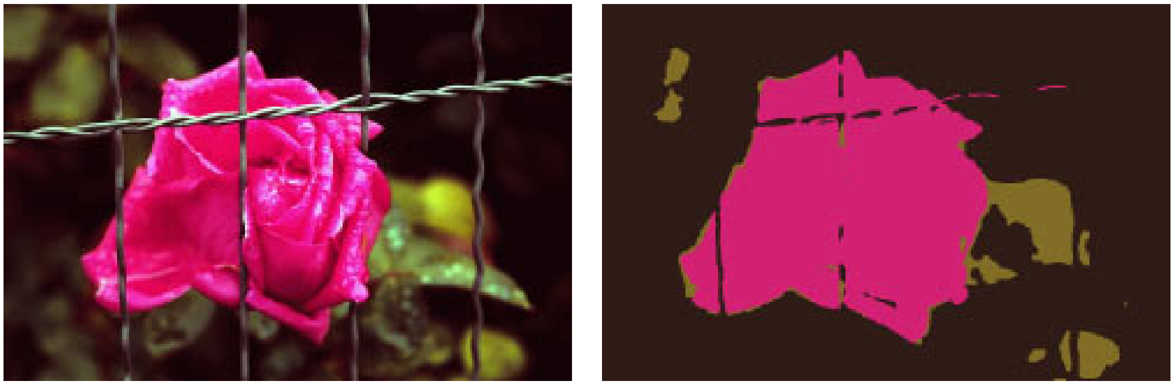
32. L. Tierney, "Markov chains for exploring posterior distributions," *The Annals of Statistics*, vol. 22, no. 4, pp. 1701–1762, 1994.

33. R. Azencott, Ed., *Parallel Simulated Annealing. Parallelization Techniques*, John Wiley & Sons, Inc., 1992.

34. Z. Kato, M. Berthod, and J. Zerubia, "Multiscale Markov random field models for parallel image classification," in *Proc. ICCV*, Berlin, May 1993.

Image name	Size	Num. classes	Estim. of $\vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma}, p$	Segm.	Estimation of num. classes	Total
LAB Color space						
peonies	256 × 256	7	26 min.	14 min.	7 min.	49 min.
flower	256 × 256	4	18 min.	9 min.	9 min.	38 min.
Carafe0	128 × 128	4	3 min.	1 min.	3 min.	9 min.
LUV Color space						
t2	432 × 666	8	117 min.	62 min.	46 min.	226 min.
flower	256 × 256	3	16 min.	8 min.	12 min.	36 min.
peonies	256 × 256	3	15 min.	8 min.	9 min.	33 min.
bird12	498 × 332	3	40 min.	20 min.	26 min.	87 min.
bleedhearts34	736 × 492	3	88 min.	44 min.	65 min.	199 min.
jellies75	217 × 404	3	19 min.	9 min.	20 min.	49 min.
seagull42	458 × 381	5	51 min.	27 min.	30 min.	109 min.
LHS Color space						
t1	497 × 502	10	130 min.	69 min.	32 min.	232 min.
peonies	256 × 256	5	21 min.	12 min.	9 min.	43 min.
bird11	498 × 332	10	80 min.	42 min.	27 min.	150 min.
bleedhearts34	736 × 492	8	130 min.	72 min.	50 min.	253 min.
kodakBus93	735 × 492	9	158 min.	84 min.	62 min.	305 min.
rose41	734 × 486	3	70 min.	29 min.	111 min.	211 min.

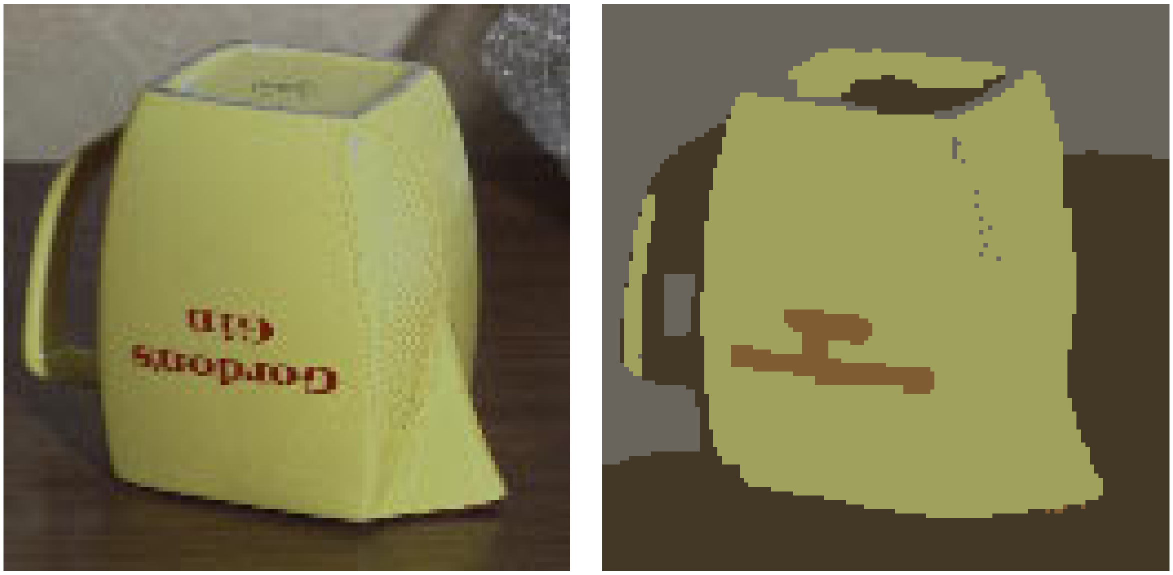
Table 1: Computing times on a Silicon Graphics Origin 2000 server. The configuration consists of 16 R10000/250MHz CPU's, each with 4 MB secondary cache, a total main memory of 8 GB, and multiple (switched) system bus bandwith's of 1.6GB/sec, up to 4GB/sec (sustained).



Original image

Segmentation result (LHS color space)

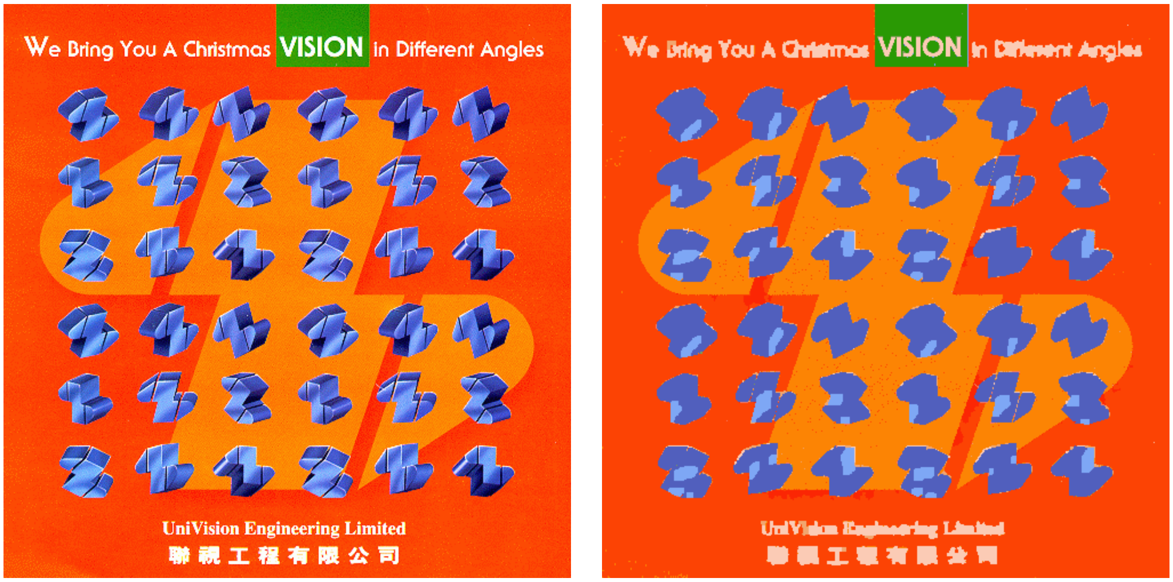
Figure 4: Segmentation of image *rose41*.



Original image

Segmentation result (LAB color space)

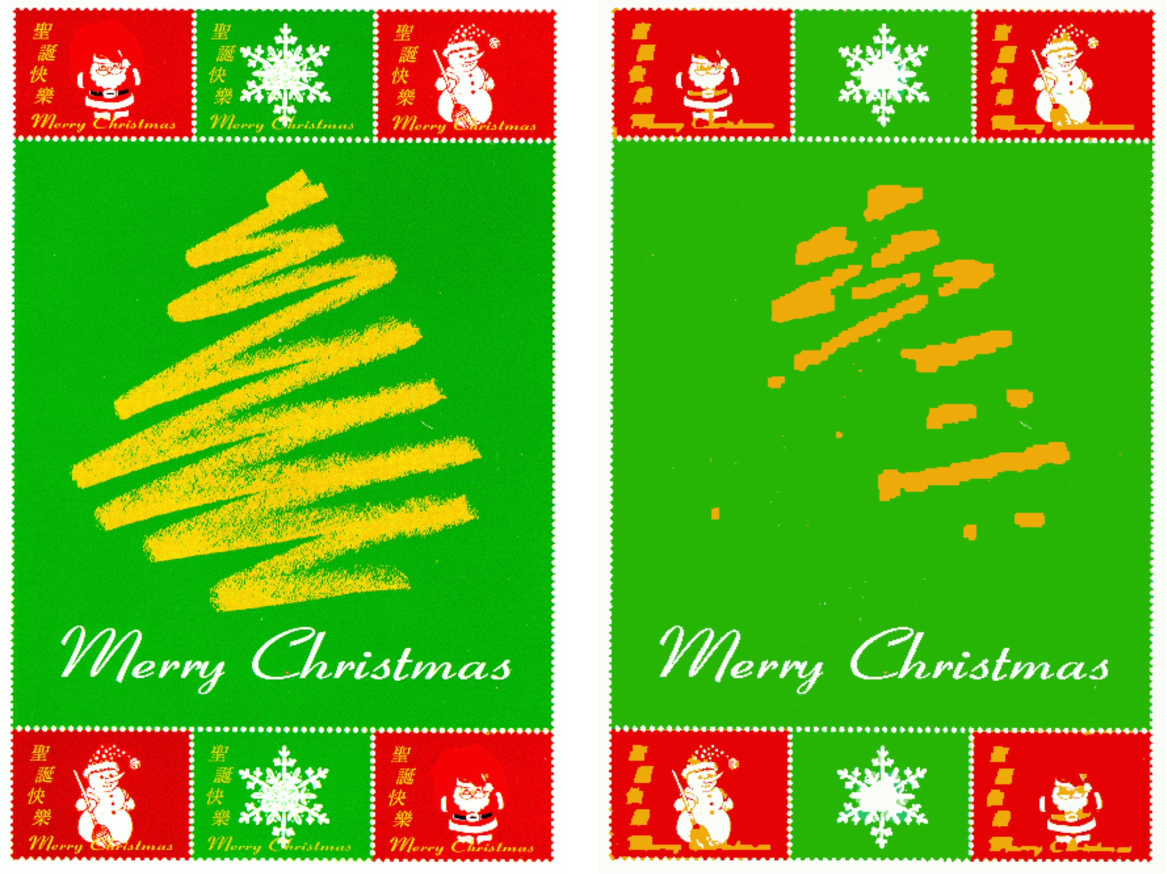
Figure 5: Segmentation of image *Carafe0*.


We Bring You A Christmas VISION in Different Angles
UniVision Engineering Limited
聯視工程有限公司

Original image

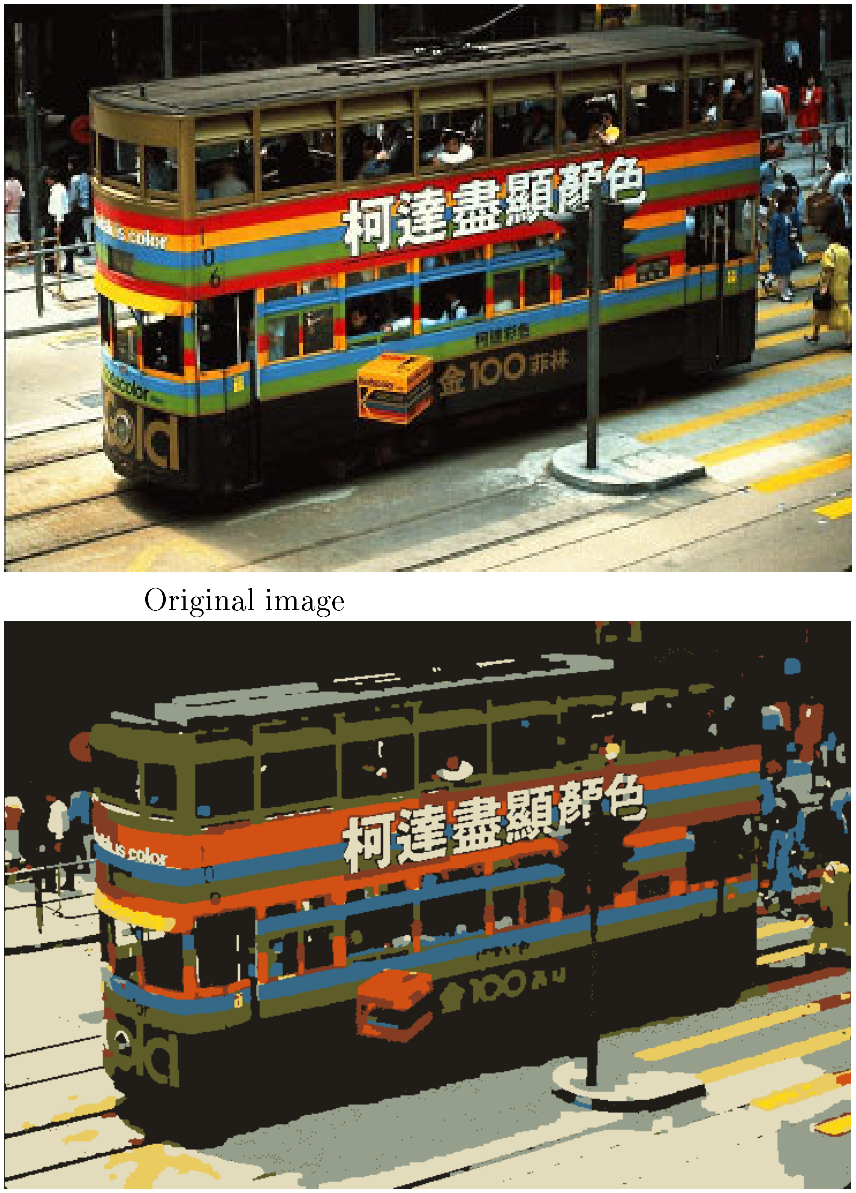
Segmentation result (LHS color space)

Figure 6: Segmentation of image *t1*.



Original image Segmentation result (LUV color space)

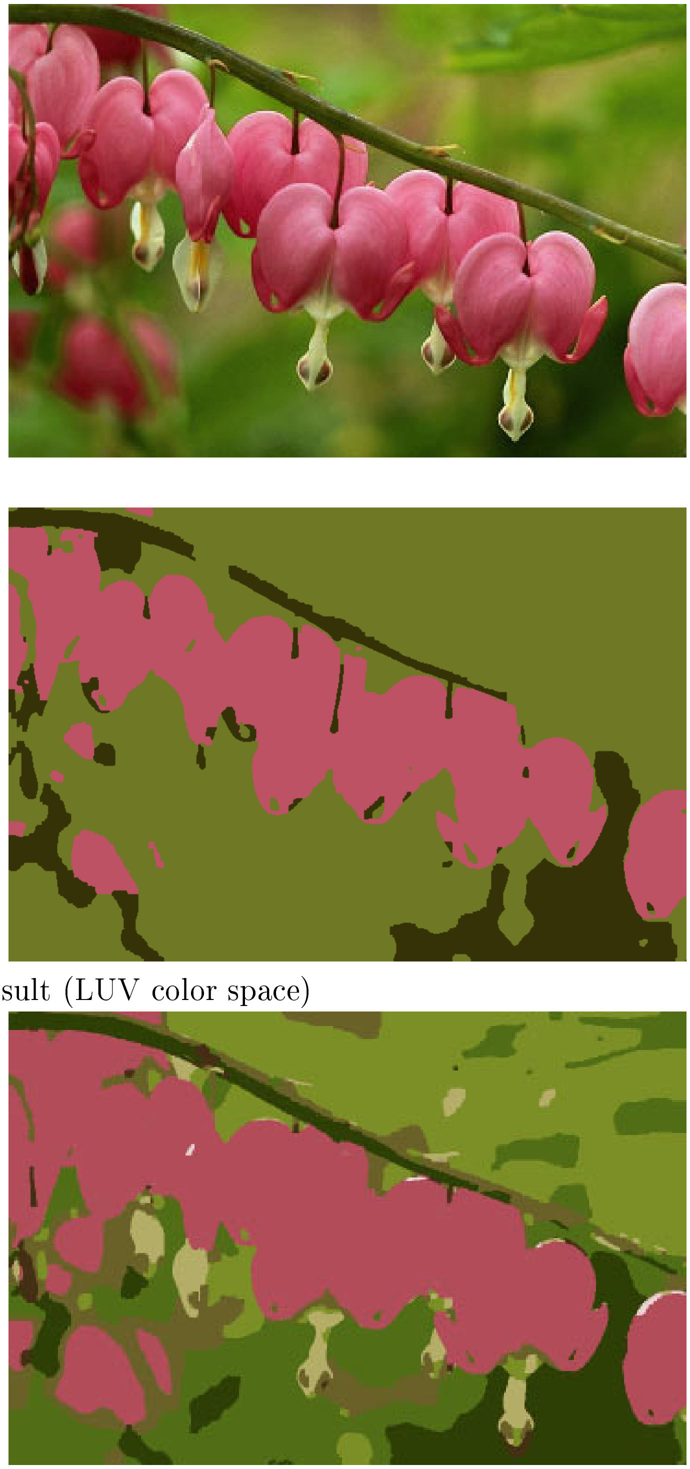
Figure 7: Segmentation of image *t2*.



Original image

Segmentation result (LHS color space)

Figure 8: Segmentation of image *kodakBus93*.

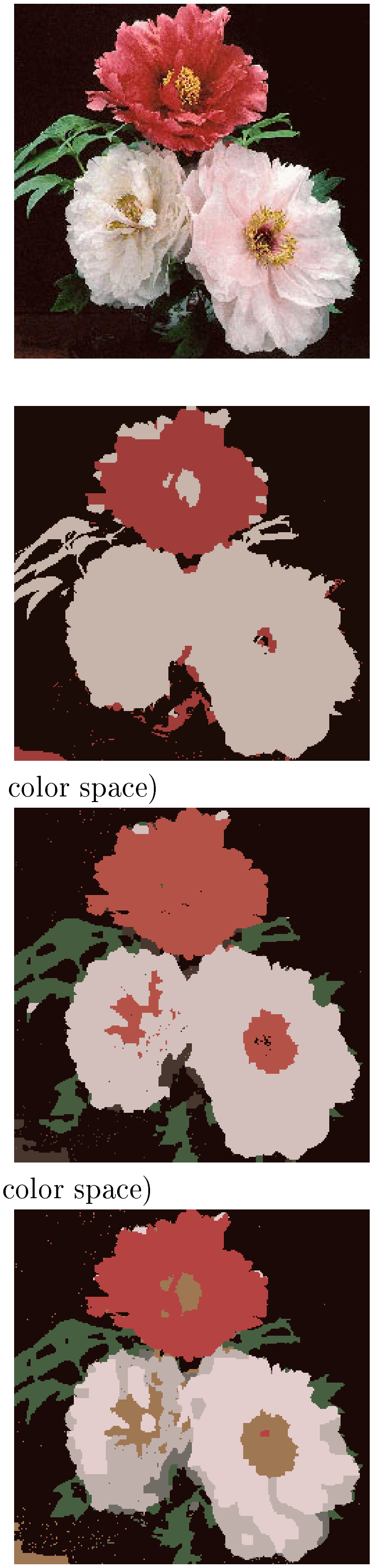


Original image

Segmentation result (LUV color space)

Segmentation result (LHS color space)

Figure 9: Segmentation of image *bleedhearts34*



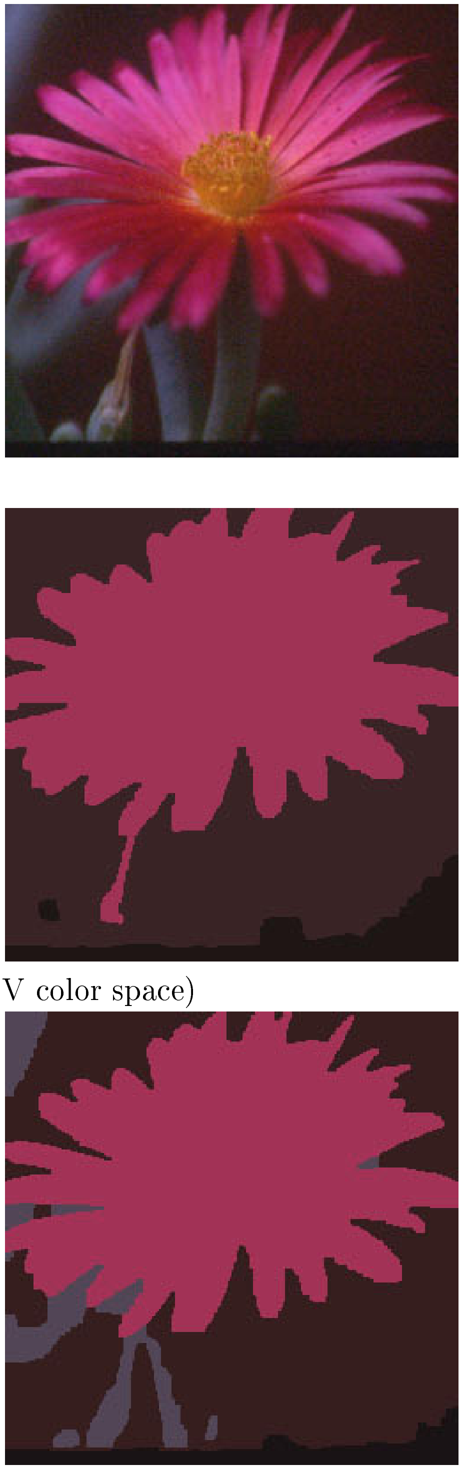
Original image

Segmentation result (LUV color space)

Segmentation result (LHS color space)

Segmentation result (LAB color space)

Figure 10: Segmentation of image *peonies*

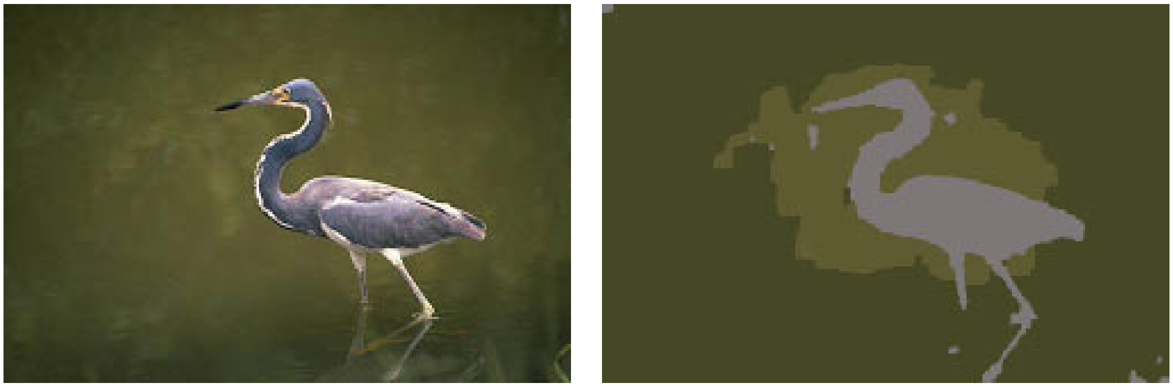


Original image

Segmentation result (LUV color space)

Segmentation result (LAB color space)

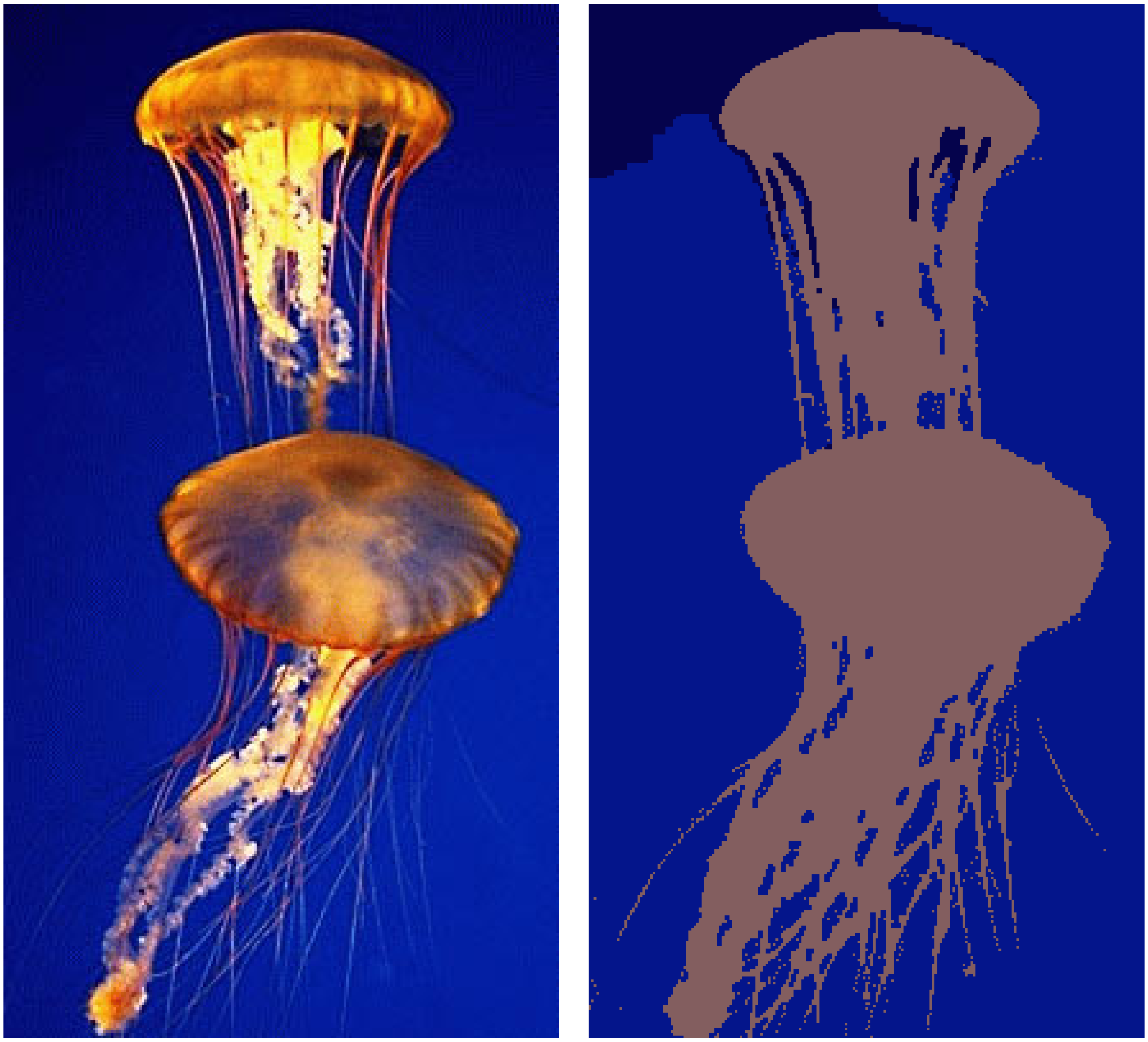
Figure 11: Segmentation of image *flower*



Original image

Segmentation result (LUV color space)

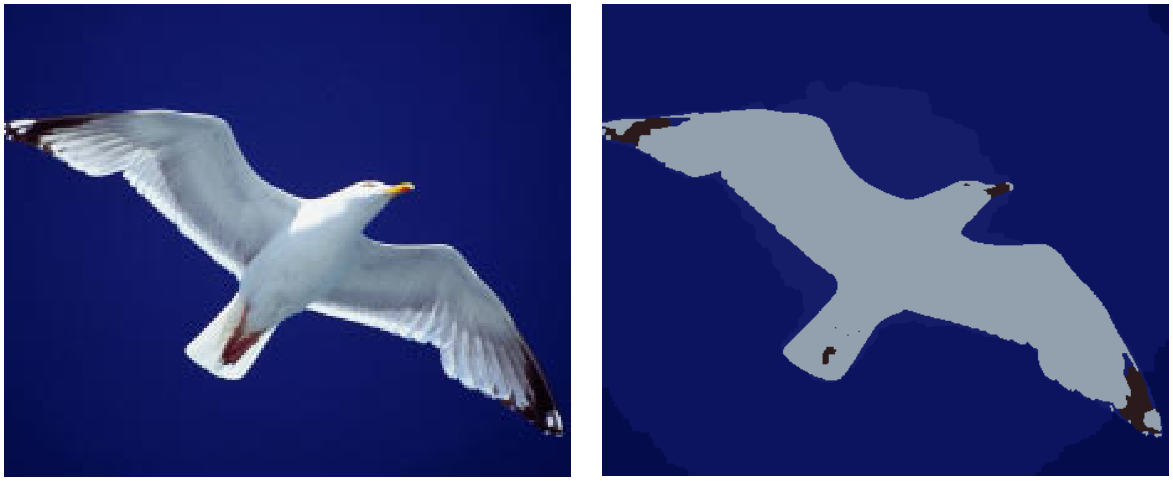
Figure 12: Segmentation of image *bird12*.



Original image

Segmentation result (LUV color space)

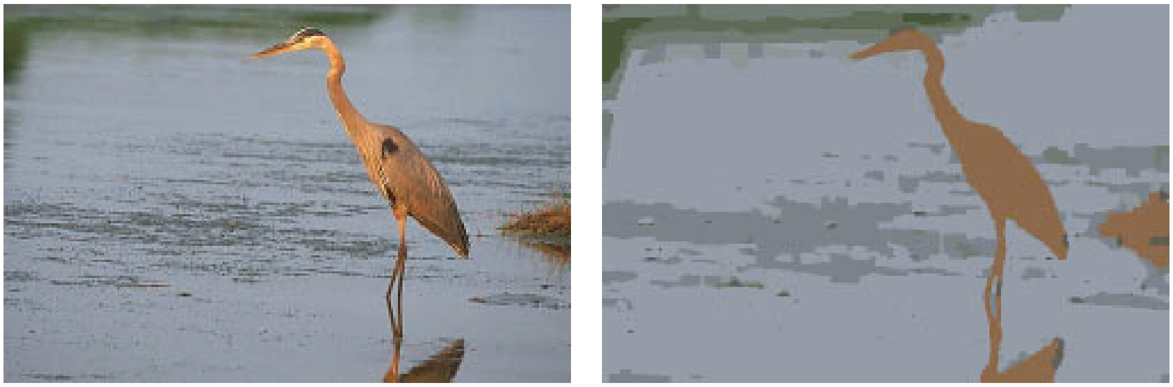
Figure 13: Segmentation of image *jellies75*.



Original image

Segmentation result (LUV color space)

Figure 14: Segmentation of image *seagull42*.



Original image

Segmentation result (LHS color space)

Figure 15: Segmentation of image *bird11*.