

# A Markov random field image segmentation model for color textured images

Zoltan Kato<sup>a,\*</sup>, Ting-Chuen Pong<sup>b,1</sup>

<sup>a</sup> Department of Informatics, University of Szeged, P.O. Box 652, H-6701 Szeged, Hungary

<sup>b</sup> Computer Science Department, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China

Received 9 February 2004; received in revised form 13 March 2006; accepted 31 March 2006

## Abstract

We propose a Markov random field (MRF) image segmentation model, which aims at combining color and texture features. The theoretical framework relies on Bayesian estimation via combinatorial optimization (simulated annealing). The segmentation is obtained by classifying the pixels into different pixel classes. These classes are represented by multi-variate Gaussian distributions. Thus, the only hypothesis about the nature of the features is that an additive Gaussian noise model is suitable to describe the feature distribution belonging to a given class. Here, we use the perceptually uniform CIE-L\*u\*v\* color values as color features and a set of Gabor filters as texture features. Gaussian parameters are either computed using a training data set or estimated from the input image. We also propose a parameter estimation method using the EM algorithm. Experimental results are provided to illustrate the performance of our method on both synthetic and natural color images.

© 2006 Elsevier B.V. All rights reserved.

**Keywords:** Segmentation; Color; Texture; Markov random fields; Parameter estimation

## 1. Introduction

Image segmentation is an important early vision task where pixels with similar features are grouped into homogeneous regions. A broadly used class of models is the so-called *cartoon model*, which has been extensively studied from both probabilistic [1] and variational [2,3] viewpoints. The model assumes that the real world scene consists of a set of regions whose observed low-level features changes slowly, but across the boundary between them, these features change abruptly. What we want to infer is a *cartoon*  $\omega$  consisting of a simplified, abstract version of the input image  $\mathcal{I}$ : regions  $R_i$  has a constant value (called a *label* in our context) and the discontinuities between them form a curve  $\Gamma$ —the contour. The pair  $(\omega, \Gamma)$  specifies a *segmentation*. Note that depending on the segmentation approach  $\Gamma$  may not be included explicitly in the model. Once  $\omega$  is determined,  $\Gamma$  is simply obtained as the discontinuities of  $\omega$ . Herein, we will also concentrate on  $\omega$ .

Taking the probabilistic approach, one usually wants to come up with a *probability measure* on the set  $\Omega$  of all

possible segmentations of  $\mathcal{I}$  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 [4,5], is to construct this probability measure in a Bayesian framework [6–8]: we shall assume that we have a set of observed ( $Y$ ) and hidden ( $X$ ) random variables. In our context, the *observation*  $\mathcal{F} \in Y$  represents the low-level features used for partitioning the image, and the hidden entity  $\omega \in X$  represents the segmentation itself. First, we have to quantify how well any occurrence of  $\omega$  fits  $\mathcal{F}$ . This is expressed by the probability distribution  $P(\mathcal{F}|\omega)$ —the *imaging model*. Second, we define a set of properties that any segmentation  $\omega$  must possess regardless the image data. These are described by  $P(\omega)$ , the *prior*, which tells us how well any occurrence  $\omega$  satisfies these properties. Factoring these distributions and applying the Bayes theorem gives us the *posterior* distribution  $P(\omega|\mathcal{F}) \propto P(\mathcal{F}|\omega)P(\omega)$ . Note that the constant factor  $1/P(\mathcal{F})$  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  $X$ .

The models of the above distributions depend also on certain parameters that we denote by  $\theta$ . Supervised segmentation assumes that these parameters are either known or a set of joint realizations of the hidden field  $X$  and observations  $Y$  (called a *training set*) is available [1,9]. In the unsupervised case, however, we know neither  $\theta$  nor  $X$ . This is called the *incomplete data* problem where both  $\theta$  and  $X$  has to be inferred from the only observable entity  $Y$ .

\* Corresponding author. Tel.: +36 62 546399; fax: +36 62 546397.

E-mail addresses: [kato@inf.u-szeged.hu](mailto:kato@inf.u-szeged.hu) (Z. Kato), [tcpong@cs.ust.hk](mailto:tcpong@cs.ust.hk) (T.-C. Pong).

<sup>1</sup> Tel.: +852 2358 7000; fax: +852 2358 1477.

*Expectation maximization* (EM) [10] and its variants (Stochastic EM [11,12], Gibbsian EM [13]), as well as *iterated conditional expectation* (ICE) [14,15] are widely used to solve such problems. It is important to note, however, that these methods calculate a local maximum [6].

### 1.1. State of the art and the proposed approach

There are many features that one can take as observation  $\mathcal{F}$  for the segmentation process: gray-level, color, motion, different texture features, etc. However, most of the segmentation algorithms presented in the literature are based on only one of these features. Recently, the segmentation of color images (textured or not) has received more attention [16–23]. In this paper, we are interested in the segmentation of color textured images. In [22], a multi-layer MRF model is proposed: each feature has its own layer, called *feature layer*, where an MRF model is defined using only the corresponding feature. A special layer is assigned to the combined MRF model. This layer interacts with both color and texture layers and provides the segmentation based on the combination of those features. Another approach is [18], where an unsupervised segmentation algorithm is proposed which uses Gaussian 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. Our approach is different in two major points. First, we use a stochastic monogrid model-based segmentation framework instead of multilayer clustering. Second, we use a combination of classical, gray-level based, texture features and color instead of direct modeling of color textures. Hence, most of the classical texture features can be used.

We have also compared our method to JSEG [21], which is a recent unsupervised segmentation algorithm for color textured images. It consists of two independent steps: First, 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. Then a region growing method is 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 [21] for details on that). Although there have been other unsupervised segmentation methods proposed (like normalized cuts [24]), we have chosen JSEG as it is also region based and uses similar cues than our method. The main difference is that JSEG is not a model-based approach.

The proposed segmentation model consists of an MRF defined over a nearest neighborhood system. The pixel classes are represented by multi-variate Gaussian distributions over image features (basically an additive Gaussian noise model). Since, the different texture-types are described by a set of Gaussian parameters, it is possible to classify or recognize textures based on prior learning of the possible parameters. Of course, if we do not have training data, parameter estimation can be a difficult task. Here, we apply the expectation–

maximization (EM) algorithm [10,25] to identify Gaussian mixture components.

In our approach, the image features  $\mathcal{F}$  consist of the perceptually uniform CIE-L\*u\*v\* color values, and texture features derived from the Gabor filtered gray-level image. Of course, the nature of the texture features is not crucial to the algorithm from the segmentation point of view. The only requirement in the current model is that Gaussian models should be suitable for describing the texture feature distributions. Most of the filtering approaches (see [26] for a comparative study of different filtering techniques) fall into this category. A detailed study about Gaussian mixture modelization of different color and texture features can be found in [27]. One can think of this kind of modelization as clustering the image pixels into different classes: The mean value represents the cluster center and the covariance matrix determines the (approximate) extent of the cluster. However, clustering algorithms work only in the feature space while our MRF segmentation model also takes into account the spatial relationship of pixels in the image domain.

## 2. Feature extraction

First, we briefly describe the color and texture features extracted from the input image  $\mathcal{I}$ . Let us assume that images are defined over a finite rectangular lattice  $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$ , where  $s = (i, j)$  denotes lattice sites (or pixels). At each pixel  $s$ , the features can be represented by a vector  $\tilde{\mathbf{f}}_s$ . The dimension of the vector is determined by the number of features extracted. Therefore, features form a vector field  $\mathcal{F} = \{\tilde{\mathbf{f}}_s | s \in \mathcal{S}\}$  (the *observation*), which is the input of our MRF segmentation algorithm discussed in Section 3.

### 2.1. Color features

The first question, when dealing with color images, is how to measure quantitatively the color difference between any two arbitrary colors. Experimental evidence suggests that the RGB tristimulus color space may be considered as a Riemannian space [28]. 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. One of these formulas is the CIE-L\*u\*v\* [28] color space that we use here. Fig. 1 demonstrates the difference between RGB and CIE-L\*u\*v\* color metrics.

### 2.2. Texture features

Many different techniques have been proposed for analyzing image texture [26]. Herein, we will adopt multi-channel filtering where the channels are represented by a bank of real-valued, even-symmetric Gabor filters. Segmentation requires simultaneous measurements in both the spatial and frequency domains. However, spatial localization of boundaries requires larger bandwidths whereas smaller bandwidths

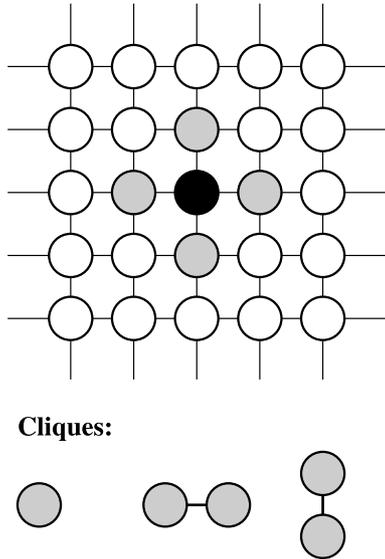


Fig. 1. First order neighborhood system with corresponding cliques.

give better texture measurements. Fortunately, Gabor filters have optimal joint localization in both domains [29]. In addition, when we are combining texture features with color, the spatial resolution is considerably increased.

The Fourier domain representation  $H(u, v)$  of the basic even-symmetric Gabor filter oriented at  $0^\circ$  is given by [29]

$$A \left( \exp \left( -\frac{1}{2} \left( \frac{(u-u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right) \right) + \exp \left( -\frac{1}{2} \left( \frac{(u+u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right) \right) \right), \quad (1)$$

where  $\sigma_u = 1/2\pi\sigma_x$ ,  $\sigma_v = 1/2\pi\sigma_y$ ,  $A = 2\sigma_x\sigma_y$ ,  $u_0$  is the frequency of a sinusoidal plane wave along the  $x$ -axis, and  $\sigma_x$  and  $\sigma_y$  are the deviations of the Gaussian envelope along the  $x$  and  $y$ -axes. Filters with other orientations can be obtained by rotating the coordinate system. In our tests, we used four orientations:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$  and the radial frequencies  $u_0$  were 1 octave apart:  $\sqrt{2}$ ,  $\sqrt{2}/2$ ,  $\sqrt{2}/4$ ,  $\sqrt{2}/8, \dots$ . For a  $2^W$  pixel wide image, the highest radial frequency falling inside the image array is  $\sqrt{2}/2^{W-2}$ .

From each filtered image  $g$ , we compute a *feature image*, which allows the segmentation of the image by its pixel values alone. As shown in Fig. 3, such a *feature image* is basically a grayscale image with similar intensities within similarly textured regions. The feature image is obtained via a local energy function applied to the filtered image. The objective of the local energy function is to estimate the energy in a local region of the filter output. The local energy function consists of a nonlinearity and smoothing (see [26] for a comparative study of different approaches). Our experiments follow the procedure reported in [29]: First, the nonlinear transformation,  $|\tanh(\alpha g_s)|$ ,  $s \in \mathcal{S}$  (a rectified sigmoid) is applied which transforms negative amplitudes to the corresponding positive amplitudes. This is followed by a Gaussian smoothing filter whose deviation is proportional to the center frequency of the Gabor filter:  $\sigma = k/$

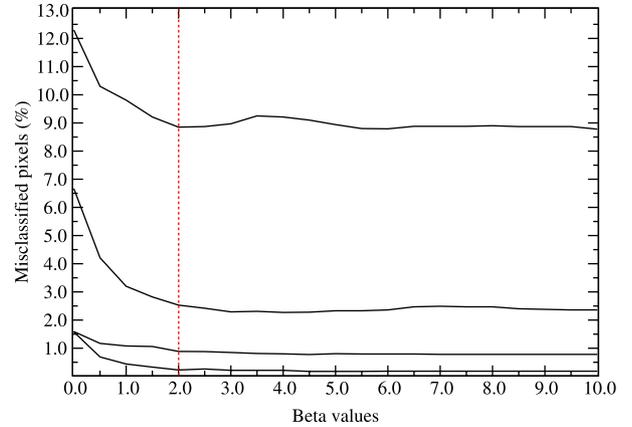


Fig. 2. Plot of the misclassification rate vs.  $\beta$  in case of supervised segmentation of the synthetic images shown in Fig. 5-Fig. 8. For  $\beta \geq 2.0$ , the error rate reaches its minimum and stabilizes.

$u_0$ . Note that accurate edge localization calls for a small Gaussian window while accurate energy estimation calls for larger window sizes. This is why the filter size is set as a function of the band center frequency. In our experiments, the Gabor filtered images are scaled to the interval  $[-1, 1]$  and we set  $\alpha = 40$  and  $k = 1$ .

### 3. MRF segmentation model

As discussed in Section 1, the segmentation is obtained as a *cartoon* image, which is basically a labeling of the input image  $\mathcal{I}$ . Hence for each pixel  $s$ , the region-type (or pixel class) that the pixel belongs to is specified by a class label,  $\omega_s$ , which is modeled as a discrete random variable taking values in  $\mathcal{L} = \{1, 2, \dots, L\}$ . The set of these labels  $\omega = \{\omega_s, s \in \mathcal{S}\}$  is a random field, called the *label process*. Furthermore, the observed image features (color and texture) are supposed to be a realization  $\mathcal{F}\{\mathbf{f}_s | s \in \mathcal{S}\}$  from another random field, which is a function of the label process  $\omega$ . Basically, the *image process*  $\mathcal{F}$  represents the manifestation of the underlying label process. Thus, the overall segmentation model is composed of the hidden label process  $\omega$  and the observable noisy image process  $\mathcal{F}$ . Our goal is to find an optimal labeling  $\hat{\omega}$  which maximizes the posterior probability  $P(\omega | \mathcal{F})$ , that is the *maximum a posteriori* (MAP) estimate [1]

$$\hat{\omega} = \arg \max_{\omega \in \Omega} P(\mathcal{F} | \omega) P(\omega), \quad (2)$$

where  $\Omega$  denotes the set of all possible labelings.

#### 3.1. Label process

The *label process*  $\omega$  is modeled as a MRF with respect to a first order neighborhood system (see Fig. 1). According to the *Hammersley–Clifford theorem* [1],  $P(\omega)$  follows a Gibbs distribution

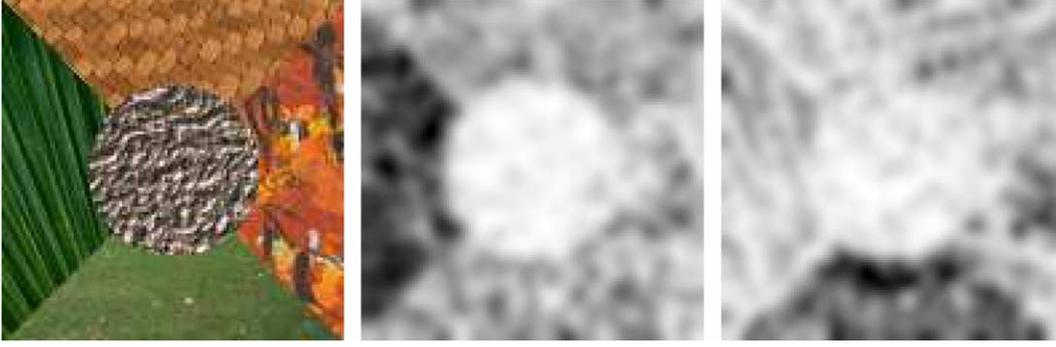


Fig. 3. Feature images derived from different Gabor filters.

$$P(\omega) = \frac{1}{Z} \exp(-U(\omega)) = \frac{1}{Z} \exp\left(-\sum_{C \in \mathcal{C}} V_C(\omega_C)\right), \quad (3)$$

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 potential* of clique  $C \in \mathcal{C}$  having the label configuration  $\omega_C$ . In our case,  $\mathcal{C}$  is the set of spatial second order cliques (*doubletons*). Each clique corresponds to a pair of neighboring pixels. The prior  $P(\omega)$  will represent the simple fact that segmentations should be locally homogeneous. Therefore, in our model, these potentials favor similar classes in neighboring pixels:

$$V_C = \delta(\omega_s, \omega_r) = \begin{cases} +1 & \text{if } \omega_s \neq \omega_r \\ -1 & \text{otherwise} \end{cases} \quad (4)$$

This potential is also known as the Potts model in statistical physics [30]. From Eq. (3), the full prior is as follows:

$$P(\omega) = \frac{1}{Z} \exp\left(-\sum_{\{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r)\right) \quad (5)$$

Note that this energy is proportional to the length of the region boundaries. Thus, homogeneous segmentations will get a higher probability.

### 3.2. Image process

The multivariate Normal density is typically an appropriate model for most classification problems where the feature vectors  $\mathbf{f}_s$  for a given class  $\lambda$  are continuous valued, mildly corrupted versions of a single mean vector  $\mu_\lambda$  [27,31]. Samples from this type of distribution tend to cluster about the mean, and the extent to which they spread out depends on the variance. This kind of modelization corresponds well to our features: Texture feature images are constructed in such a way that similar textures map to similar intensities. Hence, pixels with a given texture will be assigned a well-determined value with some variance. Furthermore, the CIE- $L^*u^*v^*$  colors form an Euclidean space [28]: pixels with similar color map to CIE- $L^*u^*v^*$  vectors that are close to their average (see Fig. 4). Putting these feature distributions into one multivariate

Normal mixture, the modes will correspond to clusters of pixels, which are homogeneous in both color and texture properties. Therefore, regions will be formed where both features are homogeneous while boundaries will be present where there is a discontinuity in either color or texture (see Fig. 7 as an example). Applying these ideas, the *image process*  $\mathcal{F}$  can be formalized as follows:  $P(\mathbf{f}_s | \omega_s)$  follows a Normal distribution  $N(\bar{\mu}, \Sigma)$ , each pixel class  $\lambda \in \mathcal{A} = \{1, 2, \dots, L\}$  is represented by its mean vector  $\bar{\mu}_\lambda$  and covariance matrix  $\Sigma_\lambda$

$$N(\mu_\lambda, \Sigma_\lambda) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_\lambda|}} \exp\left(-\frac{1}{2} (\bar{\mathbf{f}} - \bar{\mu}_\lambda) \Sigma_\lambda^{-1} (\bar{\mathbf{f}} - \bar{\mu}_\lambda)^T\right) \quad (6)$$

where  $n$  is the dimension of the combined color–texture feature space. Note that pixel features are assumed independent which yields to the following simplified form of the likelihood  $P(\mathcal{F} | \omega)$ :

$$P(\mathcal{F} | \omega) = \prod_{s \in \mathcal{S}} P(\bar{\mathbf{f}}_s | \omega_s) \quad (7)$$

$$= \prod_{s \in \mathcal{S}} \frac{1}{\sqrt{(2\pi)^n |\Sigma_{\omega_s}|}} \exp\left(-\frac{1}{2} (\bar{\mathbf{f}}_s - \bar{\mu}_{\omega_s}) \Sigma_{\omega_s}^{-1} (\bar{\mathbf{f}}_s - \bar{\mu}_{\omega_s})^T\right) \quad (8)$$

### 3.3. Posterior energy

Before calculating the posterior probability  $P(\omega | \mathcal{F})$ , we can further simplify it: first of all, the whole posterior can be expressed as a first order MRF by including the contribution of the likelihood term via the singletons (i.e. pixel sites  $s \in \mathcal{S}$ ). Indeed, the singleton energies directly reflect the probabilistic modeling of labels without context, while doubleton clique potentials express relationship between neighboring pixel labels. Therefore, after dropping the normalizing constant, we get

$$P(\omega | \mathcal{F}) \propto \exp(-U(\omega, \mathcal{F})) \quad (9)$$

$$\propto \exp\left(-\left(\sum_{s \in \mathcal{S}} V_s(\omega_s, \bar{\mathbf{f}}_s) + \beta \sum_{\{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r)\right)\right) \quad (10)$$

where  $\beta > 0$  is a weighting parameter controlling the importance of the prior. As  $\beta$  increases, the resulting regions

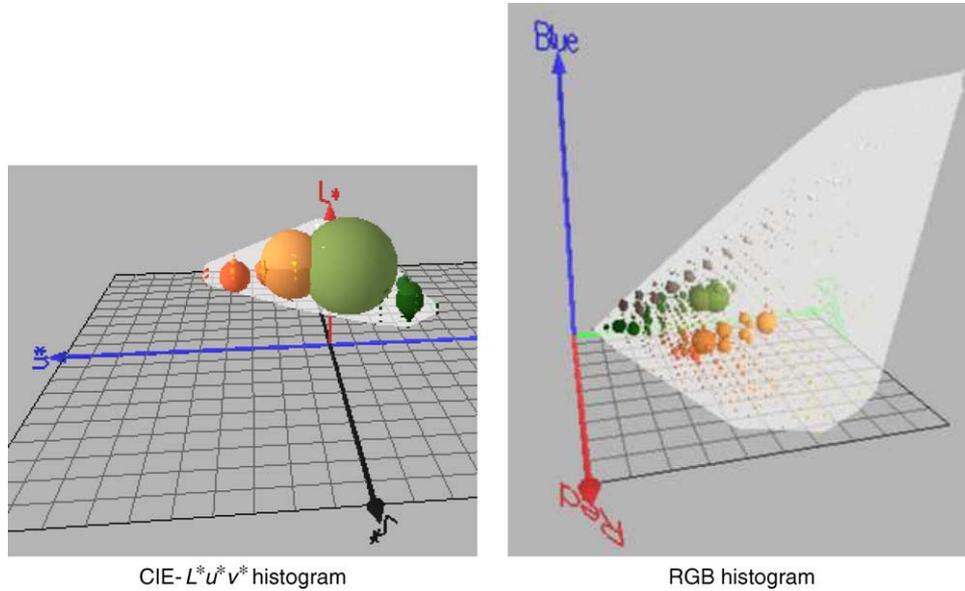


Fig. 4. Color histograms of the image in Fig. 3 generated by the ColorSpace software [39]. In CIE-L\*u\*v\* space, similar colors are well clustered but in RGB metric, these colors spread out across the whole space. This property makes the former color space a good choice when using Gaussian modeling. One can easily identify the ellipsoids corresponding to the different regions in the original image (the ellipsoid corresponding to the middle region is occluded).

become more homogeneous. The singleton potentials  $V_s(\omega_s, \vec{\mathbf{f}}_s)$  are obtained from Eq. (8):

$$V_s(\omega_s, \vec{\mathbf{f}}_s) = \ln\left(\sqrt{(2\pi)^n |\Sigma_{\omega_s}|}\right) + \frac{1}{2}(\vec{\mathbf{f}}_s - \vec{\mu}_{\omega_s}) \Sigma_{\omega_s}^{-1} (\vec{\mathbf{f}}_s - \vec{\mu}_{\omega_s})^T \quad (11)$$

Thus, the energy function  $U(\omega, \mathcal{F})$  of the so defined MRF image segmentation model has the following form:

$$\sum_{s \in \mathcal{S}} \left( \ln\left(\sqrt{(2\pi)^n |\Sigma_{\omega_s}|}\right) + \frac{1}{2}(\vec{\mathbf{f}}_s - \vec{\mu}_{\omega_s}) \Sigma_{\omega_s}^{-1} (\vec{\mathbf{f}}_s - \vec{\mu}_{\omega_s})^T \right) + \beta \sum_{\{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r) \quad (12)$$

It is clear that the original MAP estimation in Eq. (2) is equivalent to the following energy minimization problem:

$$\hat{\omega} = \arg \min_{\omega \in \Omega} U(\omega, \mathcal{F}) \quad (13)$$

Now, the segmentation problem is reduced to the minimization of the above function. Since, it is non-convex, combinatorial optimization techniques are needed to find the global minimum. Our experiments used simulated annealing with Gibbs sampler [1] and Iterated conditional modes (ICM) [32]. A comparative study of these algorithms along with other combinatorial optimization techniques applied for Bayesian image segmentation can be found in [33].

#### 4. Parameter estimation

The proposed segmentation model has the following parameters:

- (1) The weight  $\beta$  of the prior term,
- (2) the number of pixel classes  $L$ ,
- (3) the mean vector  $\vec{\mu}_\lambda$  and covariance matrix  $\Sigma_\lambda$  of each class  $\lambda \in \Lambda$ .

The automatic determination of  $L$  usually requires some additional, higher level information, which also depends on the particular application. Nevertheless, estimation of  $L$  based solely on the input image is also possible but it requires special sampling algorithms. One of these techniques is the *Reversible Jump Markov Chain Monte Carlo* algorithm, which can deal with the model dimension switching [34,35]. However, this is out of the scope of the paper. In our experiments, we simply set  $L$  manually.

While  $L$  strongly depends on the input image data,  $\beta$  is largely independent of it. Experimental evidence suggests that the model is not sensitive to a particular setting of  $\beta$  (see Fig. 2). As long as  $\beta$  is large enough, the quality of segmentations are quite similar. When  $\beta$  is close to 0, however, the prior becomes too weak and we get inhomogeneous segmentations. Although automatic estimation methods exist [36,12,37],  $\beta$  is rather a hyper-parameter, which can be fixed a priori. We found that setting  $\beta \geq 2.0$  gives satisfactory and stable segmentations.

Unlike the first two parameters, the mean and covariance of the Gaussians must be computed directly from the input image. In the remaining part of this section, we will describe how this task can be tackled.

##### 4.1. Gaussian mixture identification

Estimating the parameters of a mixture density is a well-known statistical problem. The most widely used approach is

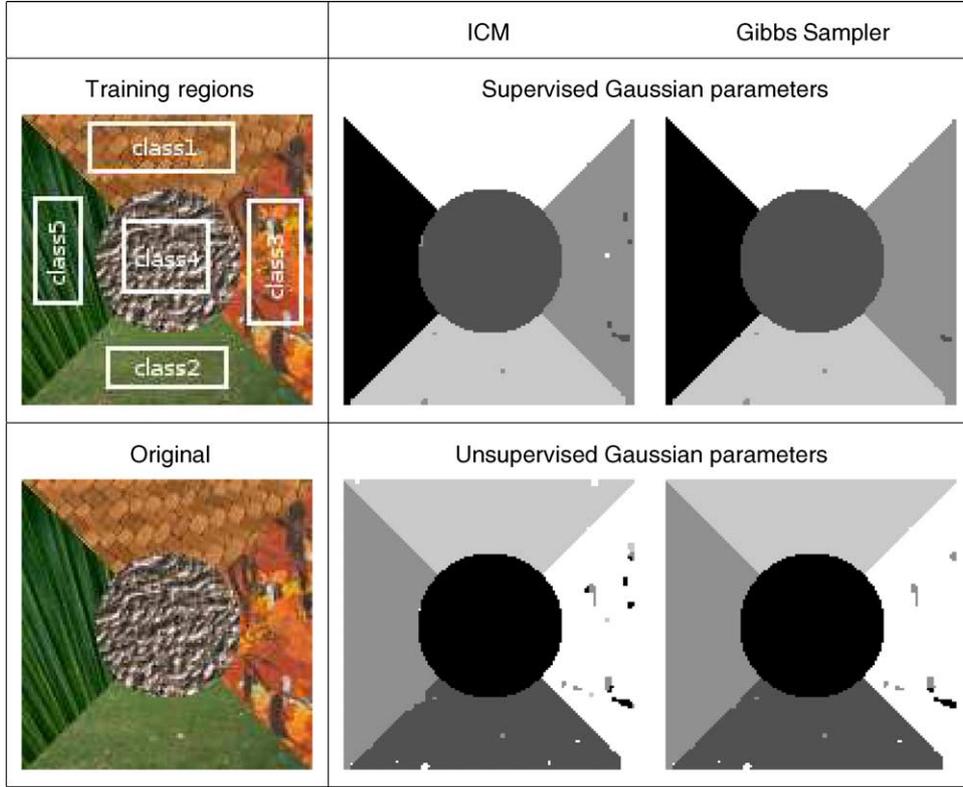


Fig. 5. Segmentation results on a  $128 \times 128$  synthetic image with 5 classes using combined features. We have used 2 Gabor filters to extract texture features. We also show the training regions used for the computation of supervised Gaussian parameters.

probably the method of *maximum likelihood*. If a labeled *training set* is available, then an unbiased estimate of  $\bar{\mu}_\lambda$  and  $\Sigma_\lambda$  can be obtained as the first and second moments of the labeled data [31]. Assuming  $\bar{\mathbf{x}}_i^\lambda (i = 1, \dots, N_\lambda)$  denotes the feature vectors assigned to class  $\lambda \in \Lambda$  and  $N_\lambda$  denotes the number of these vectors, the following formulas can be used:

$$\bar{\mu}_\lambda = \frac{1}{N_\lambda} \sum_{i=1}^{N_\lambda} \bar{\mathbf{x}}_i^\lambda \quad (14)$$

$$\Sigma_\lambda = \frac{1}{N_\lambda - 1} \sum_{i=1}^{N_\lambda} (\bar{\mathbf{x}}_i^\lambda - \bar{\mu}_\lambda)^T (\bar{\mathbf{x}}_i^\lambda - \bar{\mu}_\lambda) \quad (15)$$

Such a *training set* can be obtained by manually selecting representative regions on the input image (as shown in Fig. 5, for example). However, this procedure requires user interaction, which may not be acceptable in real life applications.

In case of incomplete data (i.e. when observations are *unlabeled*), a general iterative algorithm, known as the *EM algorithm*, has been proposed in [10] to compute the maximum likelihood estimates. Herein, we specialize this method to the mixture density context [25]. Basically, we will fit a Gaussian mixture of  $L$  components to the histogram of the image features. The observations consist of the histogram data  $\bar{\mathbf{d}}_i (i = 1, \dots, D)$  of the feature images.  $D$  denotes the number of histogram points and the dimension of a data point equals to the dimension of the combined color–texture feature space. Assuming there are  $L$  classes, we want to estimate the mean values  $\bar{\mu}_\lambda$  and covariance matrices  $\Sigma_\lambda$  for each pixel class  $\lambda \in \Lambda$ .

The *EM algorithm* aims at finding parameter values, which maximize the normalized log-likelihood function:

$$\mathcal{L} = \frac{1}{D} \sum_{i=1}^D \log \left( \sum_{\lambda \in \Lambda} P(\lambda | \bar{\mathbf{d}}_i) \right) \quad (16)$$

The underlying model is that the *complete data* includes not only the observable  $\bar{\mathbf{d}}_i$  but also the *hidden data* labels  $\bar{\ell}_i$  specifying which Gaussian process generated the data  $\bar{\mathbf{d}}_i$ . Actually,  $\bar{\ell}_i$  is also a vector of dimension  $L$  and  $\bar{\ell}_i^\lambda = 1$  if  $\bar{\mathbf{d}}_i$  belongs to class  $\lambda$  and 0 otherwise. The idea is that if labels were known, the estimation of model parameters would be equivalent to the supervised case. Hence, the following algorithm is alternating two steps: the estimation of a tentative labeling of the data followed by updating the parameter values based on the tentatively labeled data.

**Algorithm 1.** (EM for Gaussian mixture identification)

- ① [Estimation] Replace  $\bar{\ell}_i$  with its conditional expectation based on the current parameter estimates. Since, the labels may only take values 0 or 1, the expectation is basically equivalent to the posterior probability

$$P(\lambda | \bar{\mathbf{d}}_i) = \frac{P(\bar{\mathbf{d}}_i | \lambda) P(\lambda)}{\sum_{\lambda \in \Lambda} P(\bar{\mathbf{d}}_i | \lambda) P(\lambda)}, \quad (17)$$

where  $P(\lambda)$  denotes the component weight.

- ② [Maximization] Then, using the current expectation of the labels  $\bar{\ell}_i$  as the current labeling of the data, the

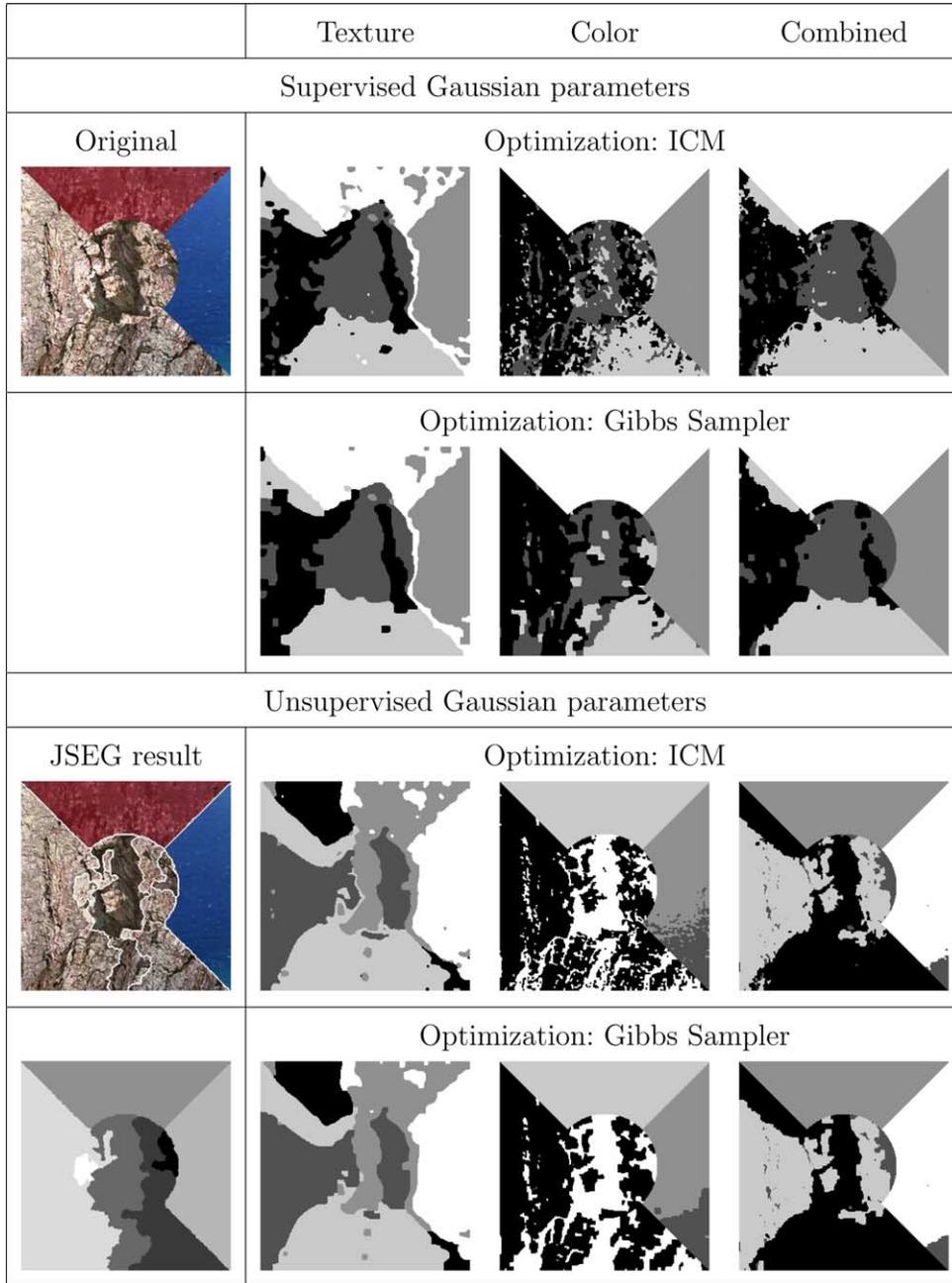


Fig. 6. Segmentation results on a  $256 \times 256$  synthetic color textured image with 5 classes. We have used 4 Gabor filters to extract texture features.

estimation of the parameters is simple

$$P(\lambda) = \frac{K_\lambda}{D} \tag{18}$$

$$\bar{\mu}_\lambda = \frac{1}{K_\lambda} \sum_{i=1}^D P(\lambda|\bar{\mathbf{d}}_i) \bar{\mathbf{d}}_i \tag{19}$$

$$\Sigma_\lambda = \frac{1}{K_\lambda} \sum_{i=1}^D P(\lambda|\bar{\mathbf{d}}_i) (\bar{\mathbf{d}}_i - \bar{\mu}_\lambda)^T (\bar{\mathbf{d}}_i - \bar{\mu}_\lambda) \tag{20}$$

where  $K_\lambda = \sum_{i=1}^D P(\lambda|\bar{\mathbf{d}}_i)$ . Basically, the posteriors  $P(\lambda|\bar{\mathbf{d}}_i)$  are used as a weight of the data vectors. They express the contribution of a particular data point  $\bar{\mathbf{d}}_i$  to the class  $\lambda$ .

- ③ Go to step ① until convergence. Each iteration is guaranteed to increase the likelihood of the estimates [10]. The algorithm is stopped when the change of the log-likelihood  $\mathcal{L}$  is less than a predetermined threshold (our test cases used  $10^{-7}$ ).

### 5. Experimental results

The proposed algorithm has been tested on a variety of color images including synthetic images (Figs. 5–8), and real (Figs. 9 and 10) scenes. We have used MIT’s VisTex database [38] to compose the synthetic color textured images. The test program has been implemented in C and run on an Intel Pentium4 3 GHz workstation. Here, we present a few examples of these

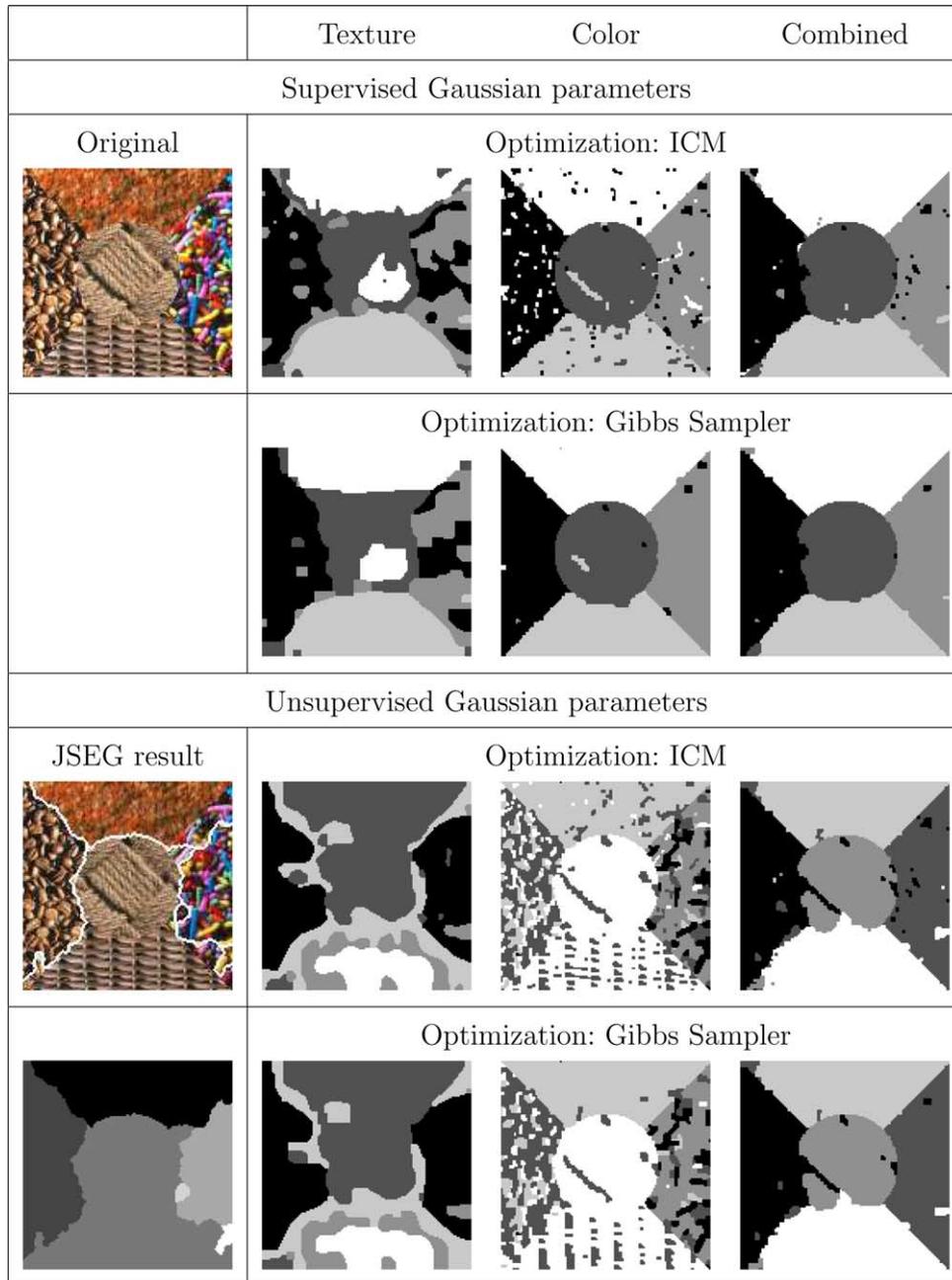


Fig. 7. Segmentation results on a  $128 \times 128$  synthetic color textured image with 5 classes. We have used 3 Gabor filters to extract texture features.

results and compare supervised and unsupervised segmentation results using color-only, texture-only and combined features. Furthermore, we also compare the results obtained via a deterministic (ICM [32]) and stochastic (simulated annealing with Gibbs sampler [1]) relaxation. For the latter algorithm, we have used an exponential annealing schedule ( $T_{k+1}=0.98T_k$ ) with an initial temperature  $T_0=4.0$ . The evaluation of the results is done in terms of both computing time and segmentation quality. The quality is measured quantitatively on the synthetic images as the percentage of misclassified pixels.

In the case of supervised segmentation, the mean vectors and covariance matrices were computed over representative

regions selected by the user (see Fig. 5). In all cases, we set  $\beta=2.5$ . This value has been found to provide satisfactory results. An optimal Gabor filter set containing 2–4 filters have been picked manually for each image. We remark, however, that it is also possible to automatically select these filters (see [26] for a collection of methods).

We found in all test cases that segmentation based purely on texture gives fuzzy boundaries but usually homogeneous regions, whereas segmentation based on color is more sensitive to local variations but provides sharp boundaries. As for the combined features, the advantages of both color and texture based segmentation have been preserved: we obtained sharp boundaries and homogeneous regions (see Fig. 5 as a good

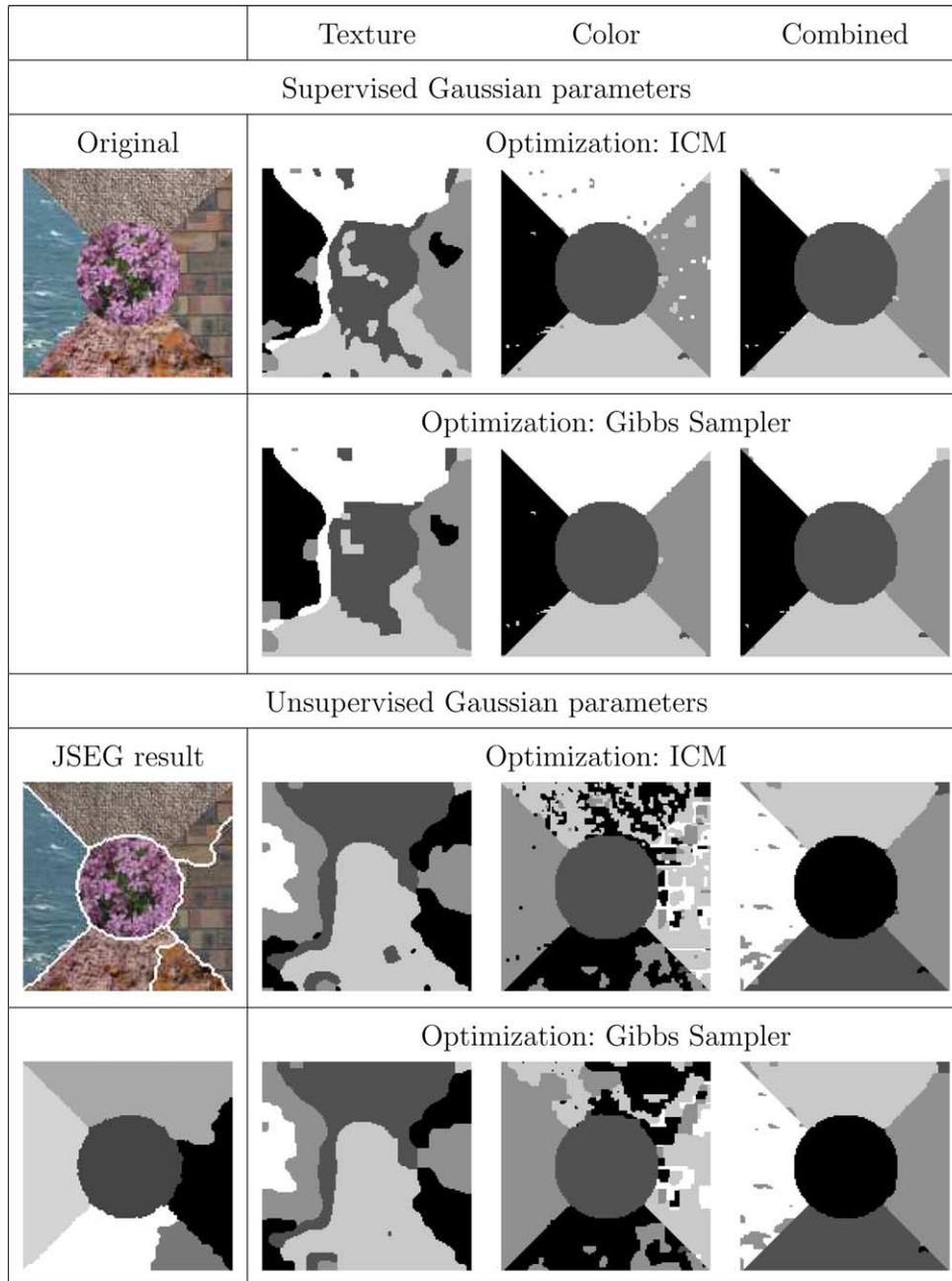


Fig. 8. Segmentation results on a  $128 \times 128$  synthetic color textured image with 5 classes. We have used 3 Gabor filters to extract texture features.

example). The power of combined features is well demonstrated by Fig. 6. Three regions contain a wooden texture with nearly matching colors and a small difference in the direction (left and lower part) or scale (middle part) in texture. The two other regions have similar texture but completely different color. Comparing color- and texture- only segmentations, the latter two regions are perfectly separated in the color segmentation but they are mixed in the texture one. On the other hand, the former three regions are much better separated in the texture segmentation than in the color one. As for the combined segmentation, the five regions are well separated, and the error rate is half of the separate segmentation rates. Of course, the quality of unsupervised segmentation is inferior to

the supervised one. Due to the difficult separability of the regions, the EM algorithm cannot provide better parameter estimates. Especially, the three regions containing the wooden texture are completely mixed, but one could argue that three classes would be more reasonable in the unsupervised case.

We can see in Table 1 that supervised, color-only segmentation provides slightly better results than combined features. However, the difference is not significant ( $\approx 0.9$ ). The reason is that color features have higher resolution, hence edges are better preserved, while the additional information provided by texture features is not enough to improve the quality of segmentation. We can observe a similar effect in Table 2, where color features alone provide good quality

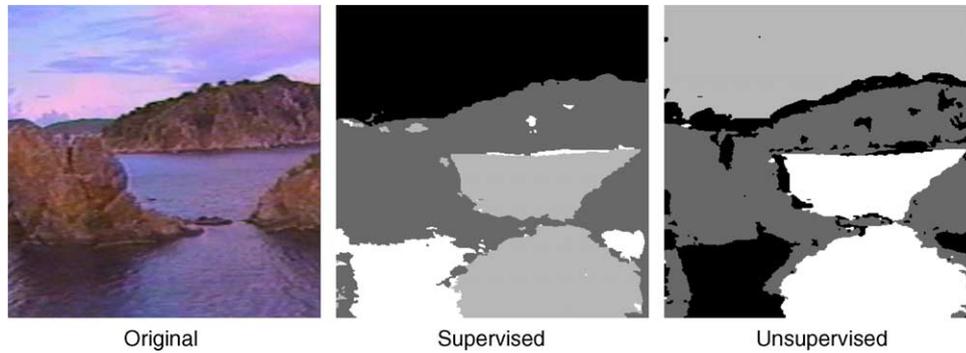


Fig. 9. Segmentation results on a  $384 \times 384$  real color image with 4 classes using combined features and ICM algorithm. We have used 3 Gabor filters to extract texture features.

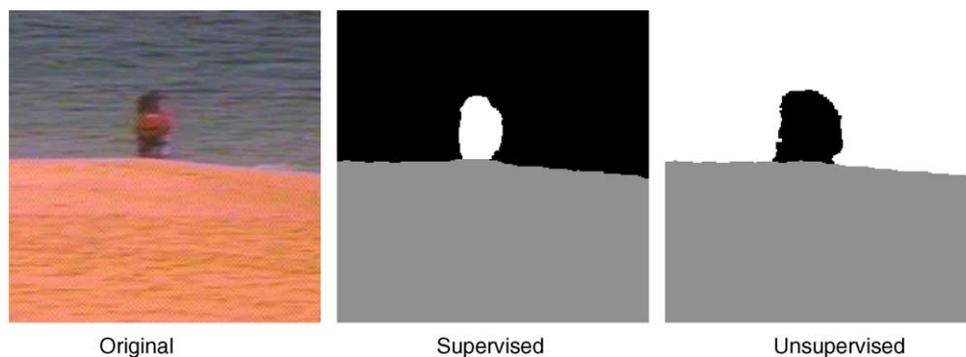


Fig. 10. Segmentation results on a  $256 \times 256$  real image with 3 classes using combined features and ICM algorithm. We have used 2 Gabor filters to extract texture features.

*supervised* segmentation when the Gibbs sampler is used. In both cases, however, the advantage of combined features is evident in the case of the ICM algorithm or unsupervised segmentation.

We found that combined features can considerably increase the quality of unsupervised segmentations. For example, three regions are merged in the color-only segmentation of Fig. 7. Since, these regions have similar colors and we have another region on the right hand side with several different colors, the EM algorithm detects only one class in these three regions and three or four classes in the region on the right hand side. As for the texture-only segmentation, the regions on the left and right hand side have similar textures thus parameter estimation tends to merge them. Similarly, the middle and upper regions are also merged. This is largely due to the fact that in the middle region,

only the finer fiber texture is detected since the region size is too small to detect the larger pattern. In case of combined features, the five regions are well separated, and results are comparable to the supervised segmentation. The findings were the same on Fig. 8.

Regarding the different optimization techniques, we can see that the suboptimal ICM provides somewhat lower quality compared to the optimal Gibbs sampler but it converges much faster (see Tables 1–4). However, this difference in quality is less important in the case of combined features and for the real images it is nearly invisible.

In Figs. 6, 7, 8, unsupervised segmentation results has also been compared to those obtained by the JSEG algorithm [21]. For producing the JSEG results, we have used the code provided by the authors and kept its default settings throughout

Table 1  
Computing times and segmentation error (misclassification rate) on a  $128 \times 128$  synthetic image (Fig. 7)

Feature	EM	Gibbs (s)	Error (%)	ICM (s)	Error (%)
<i>Supervised Gaussian parameters</i>					
Texture	N/A	8.82	22.3	0.10	25.6
Color	N/A	4.78	1.5	0.18	7.5
Combined	N/A	12.32	2.4	0.42	3.9
<i>Unsupervised Gaussian parameters</i>					
Texture	4.57 s	8.98	52.8	0.07	53.1
Color	3.94 s	6.62	36.6	0.15	41.0
Combined	7.14 s	13.34	5.85	0.34	6.8

Table 2  
Computing times and segmentation error (misclassification rate) on a  $128 \times 128$  synthetic image (Fig. 8)

Feature	EM	Gibbs (s)	Error (%)	ICM (s)	Error (%)
<i>Supervised Gaussian parameters</i>					
Texture	N/A	9.02	15.9	0.15	18.3
Color	N/A	4.49	0.6	0.20	2.0
Combined	N/A	9.42	0.9	0.28	1.0
<i>Unsupervised Gaussian parameters</i>					
Texture	4.70 s	8.39	39.8	0.13	40.7
Color	1.48 s	6.61	32.3	0.22	33.9
Combined	3.03 s	12.57	2.76	0.28	3.1

Table 3  
Computing times and segmentation error (misclassification rate) on a  $128 \times 128$  synthetic image (Fig. 5)

Feature	EM	Gibbs (s)	Error (%)	ICM (s)	Error (%)
<i>Supervised Gaussian parameters</i>					
Texture	N/A	6.25	17.5	0.07	23.3
Color	N/A	2.78	0.7	0.13	1.8
Combined	N/A	6.44	0.26	0.26	0.45
<i>Unsupervised Gaussian parameters</i>					
Texture	3.88 s	5.51	27.0	0.07	29.9
Color	2.22 s	3.82	22.6	0.15	23.9
Combined	5.46 s	7.72	0.8	0.20	1.21

Table 4  
Computing times and segmentation error (misclassification rate) on a  $256 \times 256$  synthetic image (Fig. 6)

Feature	EM	Gibbs (s)	Error (%)	ICM (s)	Error (%)
<i>Supervised Gaussian parameters</i>					
Texture	N/A	48.75	19.0	0.42	20.6
Color	N/A	33.72	19.7	0.71	23.1
Combined	N/A	91.24	8.9	2.41	11.4
<i>Unsupervised Gaussian parameters</i>					
Texture	43.48 s	45.09	39.0	0.57	39.7
Color	5.10 s	28.07	34.5	0.59	35.1
Combined	18.33 s	85.77	24.6	2.00	24.6

our test: automatic 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 and texture properties of the input image. This is why we have got more than five regions on these results. It is clear that our method outperforms JSEG. However, JSEG's advantage is that we do not have to specify the image dependent parameter  $L$ .

We also ran a test on the images published in [18] (see Figs. 9 and 10) and found that our results are comparable to those reported in [18].

## 6. Conclusions

We have proposed an MRF image segmentation model, which combines color and texture features. The model itself is not restricted to a specific texture feature. In fact, any feature is suitable as far as feature values belonging to a pixel class can be modeled by a Gaussian distribution. Due to our model-based

approach, it is also possible to classify different kind of textures based on a prior training of the corresponding parameters. The quality of the segmentation is improved with respect to color-only and texture-only segmentations. We also proposed a parameter estimation method using the EM algorithm and found that combined features can help to obtain correct estimates. Of course, unsupervised segmentation provides slightly lower quality results, but on real images, the results are comparable to supervised ones. We also tested different optimization methods and found that the suboptimal but fast ICM is a good tradeoff between quality and computing time when using combined features. Although our implementation is sequential, the segmentation algorithm is highly parallel due to the local nature of the MRF model. Thus, a parallel implementation can further improve the computing speed.

## Acknowledgements

This research was partially supported by the Janos Bolyai research fellowship of the Hungarian Academy of Sciences, the

Hungarian Scientific Research Fund-OTKA T046805, and the Hong Kong research grants DAG01/02.EG16 and HIA01/02.EG04.

## References

- [1] 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.
- [2] A. Blake, A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA, 1987.
- [3] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Communications on Pure and Applied Mathematics* 42 (1989) 577–685.
- [4] D. Kersten, P. Mamassian, A. Yuille, Object perception as Bayesian inference, *Annual Review of Psychology* 55 (2004) 271–304.
- [5] 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.
- [6] B. Chalmond, *Modeling and Inverse Problems in Image Analysis*, Springer, Berlin, 2003.
- [7] G. Winkler, *Random Fields and Markov Chain Monte Carlo Methods*, second ed., Springer, Berlin, 2003.
- [8] D. Mumford, The Bayesian rationale for energy functionals in: B. Romeny (Ed.), *Geometry-Driven Diffusion in Computer Vision*, Kluwer, Dordrecht, 1994, pp. 141–153.
- [9] N. Vandenbroucke, L. Macaire, J. Postaire, Color image segmentation by supervised pixel classification in a color texture feature space. Application to soccer image segmentation, in: *Proceedings of the International Conference on Pattern Recognition*, vol. III, Barcelona, Spain, 2000, pp. 621–624.
- [10] 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.
- [11] G. Celeux, J. Diebolt, The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem, *Computational Statistics Quarterly* 2 (1985) 73–82.
- [12] P. Masson, W. Pieczynski, SEM algorithm and unsupervised statistical segmentation of satellite images, *IEEE Transactions on Geoscience and Remote Sensing* 31 (3) (1993) 618–633.
- [13] B. Chalmond, An iterative Gibbsian technique for reconstruction of M-ary images, *Pattern Recognition* 22 (6) (1989) 747–762.
- [14] B. Braathen, W. Pieczynski, P. Masson, Global and Local Methods of Unsupervised Bayesian Segmentation of Images, *Machine Graphics and Vision* 2 (1) (1993) 39–52.
- [15] Z. Kato, M. Berthod, J. Zerubia, W. Pieczynski, Unsupervised adaptive image segmentation, in: *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 4, IEEE, Detroit, Michigan, USA, 1995, pp. 2399–2402.
- [16] 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.
- [17] J. Liu, Y.H. Yang, Multiresolution color image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (7) (1994) 689–700.
- [18] 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.
- [19] M. Mirmehdi, M. Petrou, Segmentation of color textures, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2) (2000) 142–159.
- [20] Z. Kato, T.C. Pong, A Markov random field image segmentation model using combined color and texture features in: W. Skarbek (Ed.), *Proceedings of International Conference on Computer Analysis of Images and Patterns Lecture Notes in Computer Science*, vol. 2124, Springer, Warsaw, Poland, 2001, pp. 547–554.
- [21] 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/>).
- [22] Z. Kato, T.C. Pong, G.Q. Song, Unsupervised segmentation of color textured images using a multi-layer MRF model, in: *Proceedings of International Conference on Image Processing*, vol. I, IEEE, Barcelona, Spain, 2003, pp. 961–964.
- [23] P. Nammalwar, O. Ghita, P.F. Whelan, Integration of feature distributions for colour texture segmentation in: J. Kittler, M. Petrou, M. Nixon (Eds.), *Proceedings of the International Conference on Pattern Recognition*, vol. 1, IAPR, IEEE, Cambridge, UK, 2004, pp. 716–719.
- [24] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8) (2000) 888–905.
- [25] R.A. Redner, H.F. Walker, Mixture densities, maximum likelihood and the EM algorithm, *SIAM Review* 26 (2) (1984) 195–239.
- [26] T. Randen, J.H. Husoy, Filtering for texture classification: A comparative study, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (4) (1999) 291–310 (URL <http://www.ux.his.no/~tranden/>).
- [27] 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.
- [28] S.J. Sangwine, R.E.N. Horne (Eds.), *The Colour Image Processing Handbook*, Chapman & Hall, London, 1998.
- [29] A.K. Jain, F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, *Pattern Recognition* 24 (12) (1991) 1167–1186.
- [30] R.J. Baxter, *Exactly Solved Models in Statistical Mechanics*, Academic Press, New York, 1990.
- [31] G.J. Miao, M.A. Clements, *Digital Signal Processing and Statistical Classification*, Artech House, 2002 (ISBN 1580531350).
- [32] J. Besag, On the statistical analysis of dirty pictures, *Journal of the Royal Statistical Society, Series B* 48 (3) (1986) 259–302.
- [33] M. Berthod, Z. Kato, S. Yu, J. Zerubia, Bayesian image classification using Markov random fields, *Image and Vision Computing* 14 (1996) 285–295.
- [34] S.A. Barker, P.J.W. Rayner, Unsupervised image segmentation using Markov random field models, *Pattern Recognition* 33 (4) (2000) 587–602.
- [35] Z. Kato, Reversible jump Markov chain Monte Carlo for unsupervised MRF color image segmentation in: A. Hoppe, S. Barman, T. Ellis (Eds.), *Proceedings of British Machine Vision Conference*, vol. 1, BMVA, Kingston, UK, 2004, pp. 37–46.
- [36] D. Geman, Bayesian image analysis by adaptive annealing, in: *Proceedings of International Geoscience and Remote Sensing Symposium*, IEEE, Amherst, USA, 1985, pp. 269–277.
- [37] Z. Kato, J. Zerubia, M. Berthod, Unsupervised parallel image classification using Markovian models, *Pattern Recognition* 32 (4) (1999) 591–604.
- [38] MIT VisTex, texture database, 1995 (URL <http://vismod.media.mit.edu/vismod/imagery/VisionTextur%e/vistex.html>).
- [39] P. Colantoni, ColorSpace 1.06, software, 2004 (URL <http://www.couleur.org/>).