ELSEVIER

Pattern Recognition Letters 22 (2001) 309–321

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Color image segmentation and parameter estimation in a markovian framework☆,1

Zoltan Kato [a,2], Ting-Chuen Pong [b,*], John Chung-Mong Lee [b]

[a] *Shool of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore*

[b] *Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*

Received 4 October 1999; received in revised form 13 June 2000

Abstract

An unsupervised color image segmentation algorithm is presented, using a Markov random field (MRF) pixel classification model. We propose a new method to estimate initial mean vectors effectively even if the histogram does not have clearly distinguishable peaks. The only parameter supplied by the user is the number of classes. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Unsupervised image segmentation; Color; Markov random fields; Pixel classification; Parameter estimation

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, for example) are based on such a preprocessed image. Using color information can considerably improve capabilities of image segmentation algorithms compared to purely intensity-based approaches.

In this paper, we are interested in unsupervised color pixel classification to perform image segmentation. Pixel classes are represented by multivariate Gaussian distributions and a first order Markov random field (MRF), also known as the Potts model (Baxter, 1990), is used as *a priori* model in our Bayesian classification algorithm. Some examples of MRF color image models can be found in (Daily, 1989; Huang et al., 1992; Liu and Yang, 1994). In these approaches, the color MRF process consists of the color difference of neighboring pixels. In (Daily, 1989), an MRF segmentation model with color and line processes is proposed and the use of three different lattice schemes (squares, hexagons and triangles) are discussed. The segmentation is obtained through simulated annealing. A hybrid, multi-resolution approach is presented in (Liu and Yang, 1994). First, a scale space filter (SSF) is used to determine

☆ This research was supported by Hong Kong Research Grants Council under grants *HKUST616/94E*, *HKUST661/95E* and *HKUST6072/97E*.

* Corresponding author. Tel.: +852-2358-7000; fax: +852-2358-1477.

E-mail addresses: kato@comp.nus.edu.sg (Z. Kato), tcpong@cs.ust.hk (T.-C. Pong), cmlee@cs.ust.hk (J. Chung-Mong Lee).

[1] Electronic Annexes of Figs. 7–12 available. See http://www.elsevier.nl/locate/patrec.

[2] Tel.: +65-874-8923; fax: +65-779-4580.

PII: S0167-8655(00)00106-9

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 through an MRF model defined over a quad-tree structure. The MRF model is used in (Liu and Yang, 1994) to control a split and merge algorithm. A similar model is presented in (Huang et al., 1992) but they use a monogrid MRF model and the segmentation is obtained through simulated annealing. In (Panjwani and Healey, 1995), 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.

Unsupervised segmentation algorithms are usually iterative (Geman, 1985; Lakshmanan and Derin, 1989; Kato et al., 1999), subsequently generating a labeling, estimating parameters from it, then generating a new labeling using these parameters, etc. For such a method, we need a reasonably good initial value for each parameter. Since the classes are represented by a Gaussian distribution in our model, the initialization of the mean vectors is very important because of their influence on subsequent labelings and hence on the final estimates. On the other hand, the determination of the components of a Gaussian mixture without any a priori information (Titterington et al., 1985) is a classical problem. Existing methods usually rely only on the histogram of the observed image. For noisy images where the histogram usually does not have clearly distinguishable peaks, these approaches are unreliable. We propose here a new method to find the components of such a histogram which takes into account spatial information. The basic idea is to re-quantize the observed image via a pre-segmentation. This forms the first part of an unsupervised color image segmentation algorithm. The only parameter supplied by the user is the number of classes.

In the next section, we briefly describe our MRF color image segmentation model. Then, we present an adaptive segmentation algorithm used for the estimation of the model parameters and we propose a new method to re-quantize a noisy color histogram. Finally, we present some simulation results for real and synthetic images.

2. A color image segmentation model

The first question, when dealing with color images, is how to measure quantitatively color difference between any two arbitrary colors. Experimental evidence suggests that the tristimulus color space may be considered as a Riemannian space (Jain, 1989). 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 $L^*u^*v^*$ (Jain, 1989) color space that we use herein.

The model proposed hereafter is based on an intensity based segmentation model (Kato et al., 1996). Let us suppose that the observed image $\mathcal{F} = \{\boldsymbol{f}_s | s \in \mathcal{S}\}$ consists of three spectral component values $(L^*u^*v^*)$ at each pixel $s \in \mathcal{S}$ denoted by the vector $\boldsymbol{f}_s$. We are looking for the labeling $\hat{\omega}$ which maximizes the a posteriori probability $P(\omega \mid \mathcal{F})$, that is the maximum a posteriori (MAP) estimate (Geman and Geman, 1984): arg $\max_{\omega\in\Omega} \prod_{s\in\mathcal{S}} P(\boldsymbol{f}_s \mid \omega_s))P(\omega)$, where Ω denotes the set of all possible labelings. Since our goal is to segment the image into homogeneous regions, a pixel class λ should correspond to one or more *homogeneous color patches* in the input image. Such regularities can be modelized by an additive white noise with covariance $\boldsymbol{\Sigma}_\lambda$ centered around the the expected color value μ_λ. Thus, we suppose that $P(\boldsymbol{f}_s \mid \omega_s)$ follows a Gaussian distribution and pixel classes $\lambda \in \Lambda = \{1, 2, \ldots, L\}$ are represented by the mean vectors μ_λ and the covariance matrices $\boldsymbol{\Sigma}_\lambda$. Furthermore, $P(\omega)$ is a MRF with respect to a first order neighborhood system (see Fig. 1). According to the *Hammersley–Clifford theorem* (Geman and Geman, 1984), $P(\omega)$ follows a Gibbs distribution:

$$P(\omega) = \frac{\exp(-U(\omega))}{Z(\beta)} = \frac{\prod_{C\in\mathcal{C}} \exp(-V_C(\omega_C))}{Z(\beta)}, \quad (1)$$

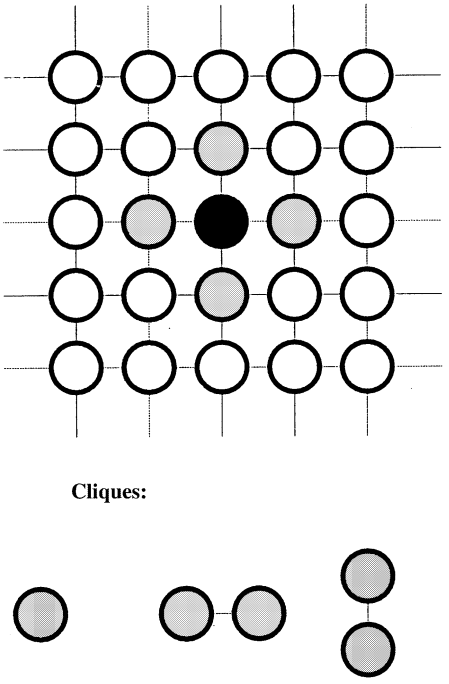


Fig. 1. First-order neighborhood system with corresponding cliques. Single pixel cliques are called *singletons*, horizontal and vertical cliques are called *doubletons*.

where $U(\omega)$ is called an *energy function*, $Z(\beta) = \sum_{\omega\in\Omega} \exp(-U(\omega))$ is the normalizing constant (or *partition function*) and V_C denotes the *clique potential* of clique $C \in \mathscr{C}$ having the label configuration ω_C. $\mathscr{C}$ is the set of spatial second order cliques (i.e., doubletons). Note that the energies of singletons (i.e., pixel sites $s \in \mathscr{S}$) directly reflect the probabilistic modeling of labels without context, while doubleton clique potentials express relationship between neighboring pixel labels. In our model, these potentials favor similar classes in neighboring pixels. Thus the energy function of the so defined MRF image segmentation model has the following form:

$$U(\omega, \mathscr{F}) = \sum_{s\in\mathscr{S}} \left(\ln\left(\sqrt{(2\pi)^3 |\mathbf{\Sigma}_{\omega_s}|}\right) + \frac{1}{2}(\boldsymbol{f}_s - \mu_{\omega_s})\mathbf{\Sigma}_{\omega_s}^{-1}(\boldsymbol{f}_s - \mu_{\omega_s})^{\mathrm{T}} \right) + \beta \sum_{\{s,r\}\in\mathscr{C}} \delta(\omega_s, \omega_r), \tag{2}$$

where $\delta(\omega_s, \omega_r) = 1$ if ω_s and ω_r are different and 0 otherwise. $\beta > 0$ is a parameter controlling the homogeneity of the regions. As β increases, the resulting regions become more homogeneous. If we know the correct parameter values (mean vectors, covariance matrices and β) then the segmentation problem, which is called *supervised* in such cases, is reduced to the minimization of the above energy function. Since it is a non-convex function, some combinatorial optimization technique is needed to tackle the problem. Herein, we will use the Metropolis algorithm (Laarhoven and Aarts, 1987) and iterated conditional modes (ICM) (Besag, 1986).

3. Parameter estimation

In real life situations, parameter values are unknown. Therefore, we need a method to estimate the parameters Θ which consists of the mean vector μ_λ and the covariance matrix $\mathbf{\Sigma}_\lambda$ for each class, and the parameter β. Since we do not have a labeled data set, we cannot use classical estimation methods such as maximum likelihood (ML). To solve this problem, we use an adaptive segmentation algorithm (Geman, 1985; Lakshmanan and Derin, 1989; Kato et al., 1999) to perform segmentation and parameter estimation simultaneously. First, let us briefly review the mathematical background of adaptive segmentation. In the case of unknown parameters, the MAP estimation problem becomes:

$$(\widehat{\omega}, \widehat{\Theta}) = \arg\max_{\omega,\Theta} P(\omega, \mathscr{F} \mid \Theta). \tag{3}$$

Since this maximization is not tractable, we use the following approximation instead (Geman, 1985; Lakshmanan and Derin, 1989):

$$\widehat{\omega} = \arg\max_{\omega} P(\omega, \mathscr{F} \mid \widehat{\Theta}), \tag{4}$$

$$\widehat{\Theta} = \arg\max_{\Theta} P(\widehat{\omega}, \mathscr{F} \mid \Theta). \tag{5}$$

Eq. (4) is simply the MAP estimate of the label field based on the estimated parameters $\widehat{\Theta}$ and Eq. (5) is the ML estimate of the parameters based on the labeled sample $(\widehat{\omega}, \mathscr{F})$. Of course, the solution of these equations is only a sub-optimal solution of the original (Eq. (3)) (Lakshmanan and

Derin, 1989). However, these equations can be solved by an iterative algorithm:

Algorithm (*Adaptive segmentation*).

1. Set $k = 0$ and initialize $\widehat{\Theta}^0$.
2. Maximize $P(\omega, \mathscr{F} \mid \widehat{\Theta}^k)$ (see Eq. (4)) using an optimization algorithm (ICM Besag, 1986, for instance). The resulting labeling is denoted by $\widehat{\omega}^{k+1}$.
3. Update the current estimate of the parameters, $\widehat{\Theta}^{k+1}$ to the ML estimate based on the current labeling $\widehat{\omega}^{k+1}$ (see Eq. (5)).
4. Goto Step (2) with $k = k + 1$ until $\widehat{\Theta}^k$ stabilizes.

Hereafter, we discuss Step (3) of this algorithm. The probability at the right-hand side of Eq. (5) can be written as $P(\widehat{\omega}, \mathscr{F} \mid \Theta) = P(\mathscr{F} \mid \widehat{\omega}, \Theta) P(\widehat{\omega} \mid \Theta)$. Using the model defined in Section 2, the first term of the above equation is a product of independent Gaussian densities and the second term is a first-order MRF. Let us now consider the log-likelihood function

$$\begin{aligned}\mathscr{L}(\Theta) &= -\ln(P(\widehat{\omega}, \mathscr{F} \mid \Theta)) \\ &= \sum_{\lambda \in \Lambda} \sum_{s \in \mathscr{S}_\lambda} \left(\ln\left(\sqrt{(2\pi)^3 |\mathbf{\Sigma}_\lambda|} \right) \right. \\ &\quad \left. + \frac{1}{2} (\boldsymbol{f}_s - \mu_\lambda) \mathbf{\Sigma}_\lambda^{-1} (\boldsymbol{f}_s - \mu_\lambda)^{\mathrm{T}} \right) \\ &\quad - \beta \sum_{\{s,r\} \in \mathscr{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r) - \ln(Z(\beta)), \end{aligned} \tag{6}$$

where $\mathscr{S}_\lambda$ is the set of pixels where $\widehat{\omega} = \lambda$. To get the minimum of the likelihood function, the derivative $\partial \mathscr{L}(\Theta)/\partial \Theta$ has to be zero at $\widehat{\Theta}$. The solution with respect to $\mu_\lambda(i)$ and $\mathbf{\Sigma}_\lambda(i,j)$ is simply the empirical mean and covariance:

$$\begin{aligned}\mu_\lambda(i) &= \frac{1}{|\mathscr{S}_\lambda|} \sum_{s \in \mathscr{S}_\lambda} \boldsymbol{f}_s(i), \ \mathbf{\Sigma}_\lambda(i,j) \\ &= \frac{1}{|\mathscr{S}_\lambda|} \sum_{s \in \mathscr{S}_\lambda} (\boldsymbol{f}_s(i) - \mu_\lambda(i)) \cdot (\boldsymbol{f}_s(j) \\ &\quad - \mu_\lambda(j)), \end{aligned} \tag{7}$$

where $|\cdot|$ denotes the cardinality. The solution with respect to β, however, is not as easy:

$$N^{\mathrm{ih}}(\widehat{\omega}) = \frac{\sum_{\omega \in \Omega} N^{\mathrm{ih}}(\omega) \exp(-\beta N^{\mathrm{ih}}(\omega))}{\sum_{\omega \in \Omega} \exp(-\beta N^{\mathrm{ih}}(\omega))}, \tag{8}$$

where $N^{\mathrm{ih}}(\widehat{\omega}) = \sum_{\{s,r\} \in \mathscr{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r)$ is the number of inhomogeneous cliques in $\widehat{\omega}$. The right-hand side is also called the *energy mean*. Since $\ln(Z(\beta))$ is *convex* in Θ (see Geman, 1985 for more details), the gradient can be approximated by stochastic relaxation. Herein, we use a simpler heuristic, which is computationally less expensive and gives reasonably good results (Kato et al., 1999). Suppose that we have an estimate of the label field ($\widehat{\omega}$). The algorithm aims at finding a $\widehat{\beta}$ which does not change the labeling $\widehat{\omega}$ during a few iterations of a fixed temperature, T, Metropolis algorithm. Basically, the algorithm uses a trial and error strategy where the change of the current β value is governed by the ratio of the number of inhomogeneous cliques of the current labeling $\widehat{\omega}$ and the one generated by the Metropolis algorithm (see Kato et al., 1999 for more details). T is chosen empirically on a trial and error basis. In our tests, we have set $T = 2.5$. This value has proven successful for intensity-based segmentation (Kato et al., 1999). The idea behind $T = 2.5$ is that an excessively large value ($T \geqslant 4$) would result in a completely random labeling independent of $\widehat{\omega}$ and the algorithm will not converge. On the other hand, an excessively small value ($T \leqslant 1$) turns the Metropolis algorithm into a deterministic one, which permits a large variation in $\widehat{\beta}$ without really disturbing $\widehat{\omega}$.

4. Obtaining initial parameters

The estimation procedure described previously supposes that we have an initial guess about the parameters. Experiments show that the most crucial value is the mean value, all the other parameters are far less sensitive to initialization. In Figs. 2 and 3, we can see that the quality of the final segmentation is not influenced by the initialization of β. We have got a similar result for the covariance matrix in Fig. 3. However, we found that slightly changing the initial mean values may cause a significant degradation in the final result (see Fig. 3).

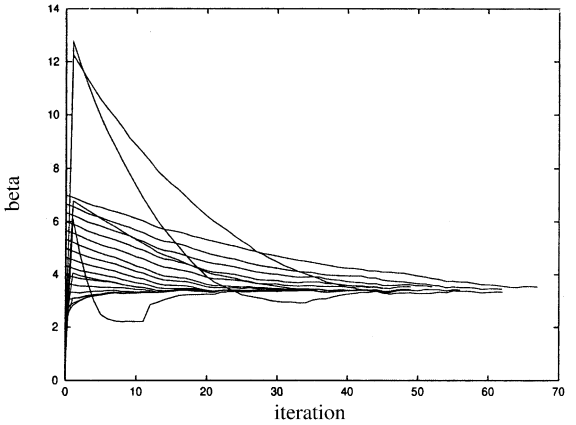


Fig. 2. Convergence of the β parameter in case of the *synthetic* image shown in Fig. 9. The initial values were $0.33, 0.66, \ldots, 6.99$. All other parameters are set to the correct (supervised) values (see Table 1). The final estimate is obtained between 3.3 and 3.5, whereas the supervised value was $\beta = 3.5$.

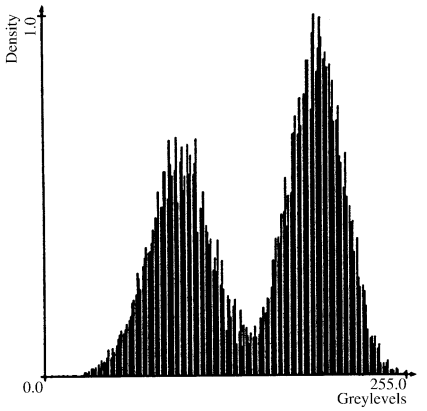


Fig. 4. Histogram of the *synthetic* image's v^* component.

Thus the main problem is to obtain initial mean values as close as possible to the correct ones.

Estimating the mean values is a classical problem, namely the determination of the components of a Gaussian mixture without any a priori information. Unfortunately, classical methods (Titterington et al., 1985) fail if the histogram does not have clearly distinguishable peaks, which is often the case in dealing with noisy images. For example, in Fig. 4, we show the histogram of a noisy synthetic image (SNR = 5 dB). Any histogram-based method will fail to find a reasonably good mean value for the four classes we have in this image. However, using our segmentation-based initialization method, we are able to obtain the histogram shown in Fig. 5.

Another problem, specific to color images is a sparse histogram. Due to the large number of possible colors ($256^3 = 16777216$ in case of 24 bit color codes), typically less than 10 pixels belong to the same color value yielding a completely flat

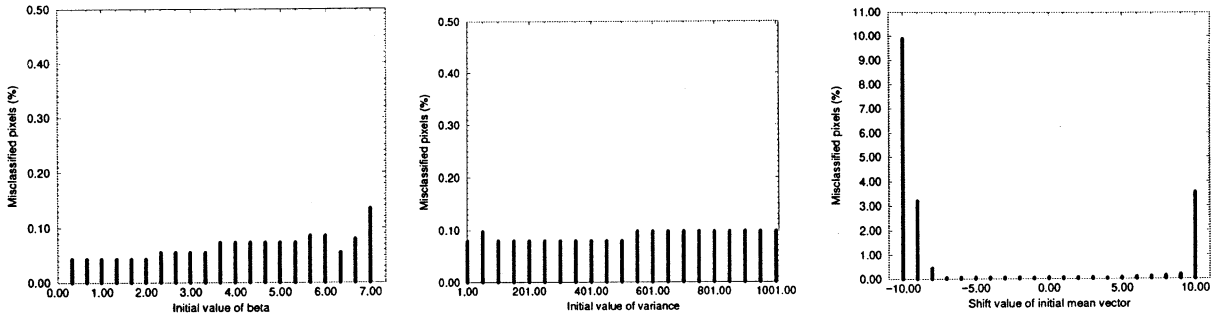


Fig. 3. Plots of the percentage of misclassified pixels versus the initial value of β, variance and mean value in case of the *synthetic* image (Fig. 9). During the experiments, only the tested parameter is changed, all other parameters are set to their supervised value (see Table 1). For β, the ratio of misclassified pixels is well under 0.2%. We have got a similar result for the covariance matrix $\mathbf{\Sigma}$. The initial covariance matrix is diagonal with variances set to $1, 51, 101, \ldots, 1001$. Independent of the initialization, we have obtained final estimates close to the supervised values shown in Table 1. The ratio of misclassified pixels is well under 0.2%. In case of the mean values, however, we found that using initial values obtained by slightly changing the correct mean values (they were shifted by $-10, -9, \ldots, 0, \ldots, 9, 10$) causes an important degradation in the final result and the algorithm is unable to find the supervised values shown in Table 1. We can see that the ratio of misclassified pixels reaches 10%.

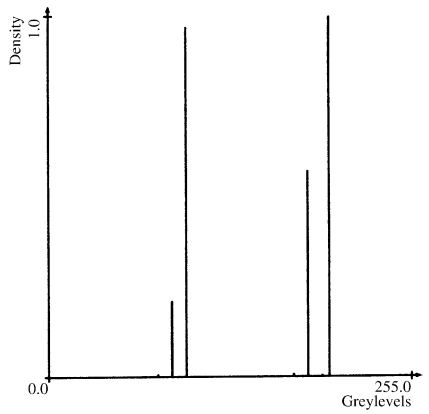


Fig. 5. Histogram of the *synthetic* image's v^* component after pre-segmentation and quantization.

histogram. As a result, the histogram of such an image will be too sparse for statistical analysis. To solve this problem, we re-quantize our image. Our goal is to reduce the number of colors such that the quantized image contains only those colors which correspond to important regions in the original image. However, the goal of a classical quantization algorithm is the minimization of the *perceived difference* between the original and quantized images (Jain, 1989; Verevka and Buchanan, 1995). Hence, using a classical quantization algorithm could result in losing important spatial information (merging color values that correspond to different spatial regions in the input image). Thus, we have to re-quantize the image taking into account *spatial information*. In (Verevka and Buchanan, 1995), a clustering algorithm (called LKM) based on the *K*-means algorithm is proposed. For comparison, we present the results obtained on the *postcard* image (see Fig. 6). Although the results of the LKM algorithm are a perceptually better reproduction of the original image, the resulting initial mean values are not suitable for the segmentation algorithm. Hence, inferior final segmentation results are obtained.

Original postcard image.

Results after quantization via the LKM algorithm colored by random colors.

Final unsupervised segmentation.

Fig. 6. Quantization and final result obtained from a *postcard* image using the LKM quantization algorithm. The quantized image is closer to the original image but its histogram is not suitable to obtain initial mean values for the segmentation process. Hence, important regions are lost on the final segmentation.

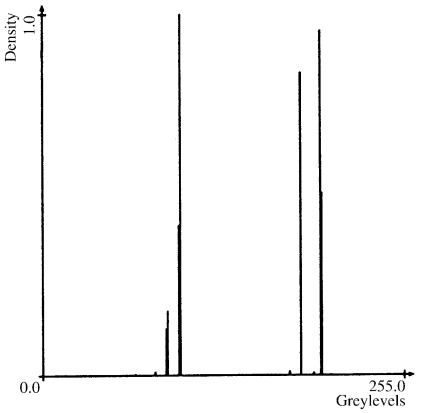


Fig. 7. Histogram of the *synthetic* image's v^* component after pre-segmentation.

We now introduce our approach. It is based on a pre-segmentation. Instead of analyzing the histogram of the observed image, an initial segmentation is obtained via a split and merge algorithm using color difference as a measure of homogeneity. In the $L^*u^*v^*$ color space, a perceptually relevant color difference is obtained from the second norm (i.e., Euclidean distance) of two color vectors. Regions are represented by the average color of the original pixels. Two neighboring regions are merged if their color difference is less than a certain threshold τ. A region is split if its maximal color difference is larger than τ. A small τ results in a large quantity of small regions, a large τ gives a small quantity of large regions. To keep the method unsupervised, we have to determine τ from the observed image. This is easily achieved because we only require a *reasonably* good initial segmentation, thus the number and size of regions is not crucial. In our tests, we have found that a τ, obtained as 25% of the maximal color difference in the observed image, gives reasonably good pre-segmentation.

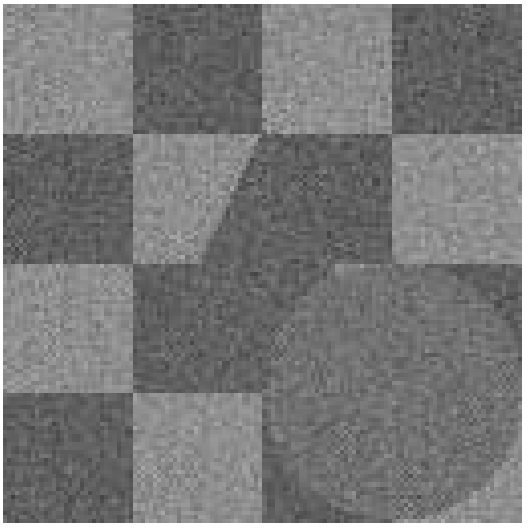
Original postcard image.

Result after pre-segmentation and quantization colored by random colors.

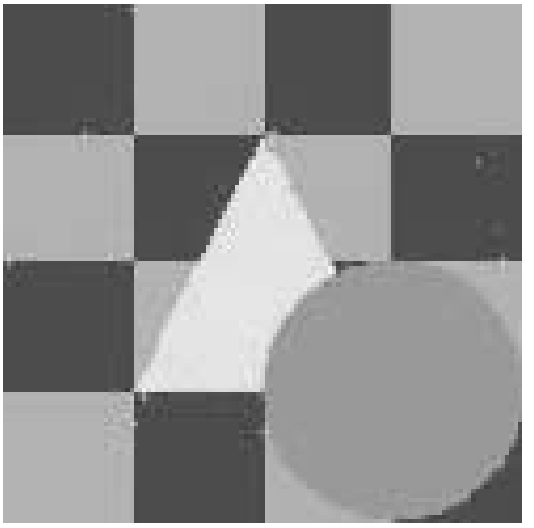
Final unsupervised segmentation.

Fig. 8. Pre-segmentation and final result obtained from a *postcard* image. On the pre-segmentation result, we can see a yellow ($L^*u^*v^* \approx [220, 80, 240]$) and a green ($L^*u^*v^* \approx [160, 30, 190]$) regions in the central part of the image. Since these colors are close to each other (the difference is only 92 while $\tau = 100$ for this particular image), the split and merge algorithm has a tendency to merge them. In addition, there is no clear boundary between these regions which increase the probability of the formation of mixed regions. This is why we obtain two large regions, one of them close to yellow ($L^*u^*v^* \approx [196, 47, 204]$) and the other one close to green ($L^*u^*v^* \approx [172, 34, 190]$). Although the pre-segmentation is quite coarse, its histogram contains the colors of all important regions. Thus it provides reasonably good initial mean values which was the goal of pre-segmentation.

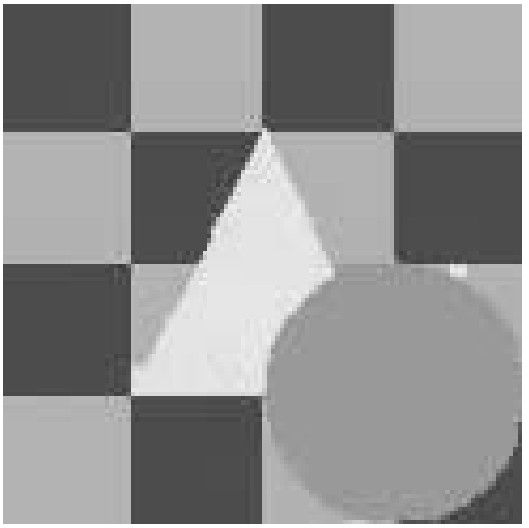
The histogram of the pre-segmented image has clear peaks, it is suitable now for statistical analysis. In Fig. 7, we can observe that the histogram has only a few peaks, some of them very close to each other. Since in the pre-segmented image, the pixel colors have been replaced by the average color of their region, we expect to get the most significant peaks close to the mean value of the pixel classes. Thus, we could simply select the L (L denotes the number of pixel classes) most significant peaks. Unfortunately, the pre-segmentation provides regions having slightly different average color. This might result in more than one peak corresponding to the same pixel class and



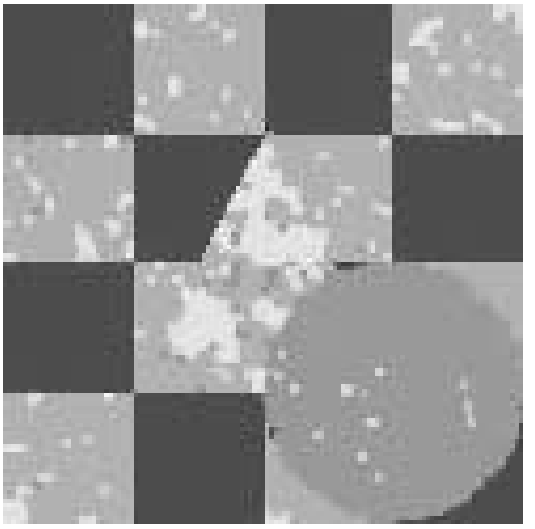
Noisy synthetic image (SNR=5db).



Supervised segmentation result using color information (0.05% misclassified pixels).



Unsupervised segmentation result using color information (0.19% misclassified pixels).



Supervised segmentation result using only intensity information (9.19% misclassified pixels).

Fig. 9. Supervised and unsupervised segmentation obtained from a 128×128 noisy *synthetic* image. The signal to noise ratio (SNR) was 5 dB in the RGB image.

cause detection of false mean values which could lead to wrong initial estimations. In order to suppress these unnecessary peaks, we merge neighboring peaks by quantizing the pre-segmented image. Experiments show that a 20% reduction of the number of colors is sufficient (see Fig. 5) to remove these peaks.

From the histogram obtained by the pre-segmentation and quantization process, we can easily extract initial mean vectors as the coordinates of the first L most significant peaks. At this stage, we only use the 2D histogram of the u^*v^* space because these components carry the chromatic information. As a consequence, the L^* component is omitted in the first step of the *Adaptive segmentation* algorithm, but it is computed from the first segmentation and is used in subsequent iterations.

It may seem strange to use a segmentation algorithm to get initial parameters for a pixel classification algorithm. Why not use the result of the pre-segmentation? To clarify this issue, consider the image in Fig. 8. It contains large homogeneous regions but also small, fine details. Since the pre-segmentation algorithm uses only the color difference, it is fast but unable to keep fine details. In particular, the central region of the *postcard* image has been segmented into two regions producing a segmentation that is not evident in the original image. This is a good example to demonstrate that a coarse pre-segmentation can be viable when it comes to re-quantizing the image. The final, good quality result is then obtained by the more elaborate MRF model. We remark, that we usually get better pre-segmentations. Thus consider Fig. 8 as an extreme example.

5. Experimental results

The proposed algorithm has been tested on a variety of images including synthetic noisy images, outdoor and indoor scenes, and video sequences. The test program has been implemented in C. Herein, we present a few examples of these results and compare color- and intensity-based segmentation as well as supervised and unsupervised results. In the case of supervised segmentation, the mean vectors and covariance matrices were computed over representative regions selected by the user and β was set on an ad hoc basis. In both cases, the number of pixel-classes is given by the user. In our experiments, a reason-

Table 1
Parameter values of the *synthetic* image[a]

Class	Initial values		Final estimates		Supervised values	
#	μ	Σ	μ	Σ	μ	Σ
1	(–, 42, 197)	I	(189, 41, 197)	$\begin{pmatrix} 280 & -155 & 230 \\ -155 & 137 & -126 \\ 230 & -126 & 312 \end{pmatrix}$	(189, 41, 197)	$\begin{pmatrix} 280 & -155 & 230 \\ -155 & 137 & -126 \\ 230 & -126 & 312 \end{pmatrix}$
2	(–, 82, 97)	I	(96, 81, 97)	$\begin{pmatrix} 474 & -259 & 419 \\ -259 & 209 & -224 \\ 419 & -224 & 500 \end{pmatrix}$	(96, 81, 97)	$\begin{pmatrix} 474 & -260 & 420 \\ -260 & 209 & -225 \\ 420 & -225 & 501 \end{pmatrix}$
3	(–, 97, 87)	I	(104, 98, 87)	$\begin{pmatrix} 438 & -246 & 386 \\ -246 & 203 & -215 \\ 386 & -215 & 456 \end{pmatrix}$	(105, 98, 87)	$\begin{pmatrix} 429 & -248 & 385 \\ -248 & 202 & -215 \\ 385 & -215 & 456 \end{pmatrix}$
4	(–, 47, 182)	I	(132, 49, 181)	$\begin{pmatrix} 366 & -197 & 295 \\ -197 & 179 & -160 \\ 295 & -160 & 414 \end{pmatrix}$	(132, 49, 181)	$\begin{pmatrix} 366 & -197 & 294 \\ -197 & 179 & -159 \\ 294 & -159 & 412 \end{pmatrix}$
β	2.0		3.5		3.5	

[a] The table shows the initial parameter values, final estimates obtained trough *Adaptive segmentation* and the correct, supervised values. One can see, that initial mean values are already close to the correct ones but we have used a unit matrix (denoted by I in the table) as initial value for the covariance matrices. Final estimates are nearly the same as supervised parameters.

able value has been chosen depending on how many differently colored regions can be found in the image. Looking at the histogram of the image is another clue when deciding about the number of pixel-classes. Regions with similar colors tend to merge during segmentation yielding to a random mixture of the corresponding pixel classes. Thus classes with similar mean-values should be avoided.

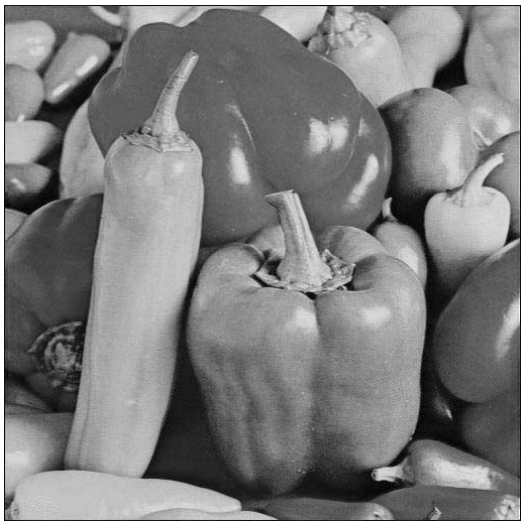
First, the original images were converted from RGB to $L^*u^*v^*$ using the equations from (Jain, 1989). The dynamic range of all color components was [0, 255]. Then, we applied the split and merge algorithm followed by a quantization and the initial mean values were extracted from the obtained image as described in Section 4. Independent of the input image, the initial covariance matrix was set to the unit matrix and the initial $\beta = 2.0$. The final parameters and segmentation are then obtained through the *Adaptive segmentation* algorithm.

In Fig. 9, we show the results obtained from a noisy color synthetic image by *supervised* segmentation. Comparing color- and intensity based segmentation, it is clear that using color information can significantly improve the final result (0.05%

Table 2
Computing times on a Silicon Graphics Origin 2000 server[a]

Image	Image size	Number of classes	Pre-segmentation	Quantization (s)	MRF segmentation	Total
Synthetic	128×128	4	4.91 s	0.03	54 s	58.94 s
Postcard	432×666	5	10 m 45 s	0.45	11 m 56 s	22 m 41 s
Peppers	512×512	2	5 m 14 s	0.39	11 m 48 s	17 m 2 s
Jellies	217×404	2	49.8 s	0.15	3 m 35 s	4 m 25 s
Seagull	458×381	2	58.3 s	0.27	9 m 42 s	10 m 40 s
Leaves	738×490	5	26 m 15 s	0.55	31 m 57 s	58 m 12 s

[a] The configuration consists of 16 R10000/250 MHz CPU's, each with 4 MB secondary cache, a total main memory of 8 GB, and multiple (switched) system bus bandwith's of 1.6 GB/s, up to 4 GB/s (sustained).



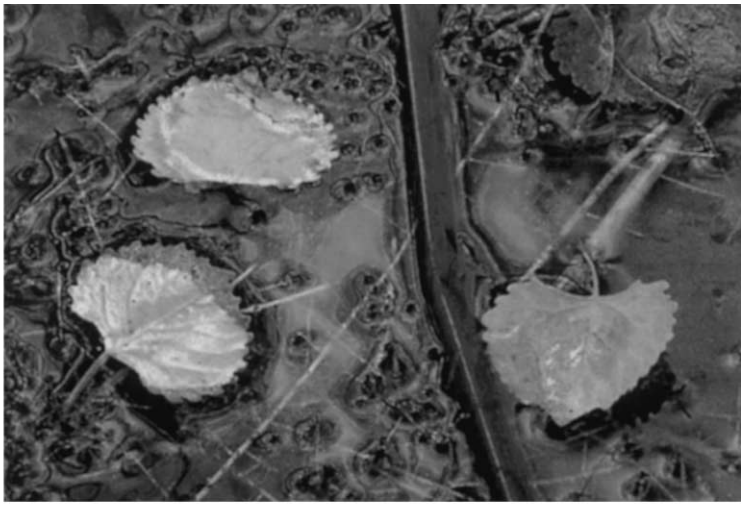
Original image (peppers).

Unsupervised segmentation result (2 classes).

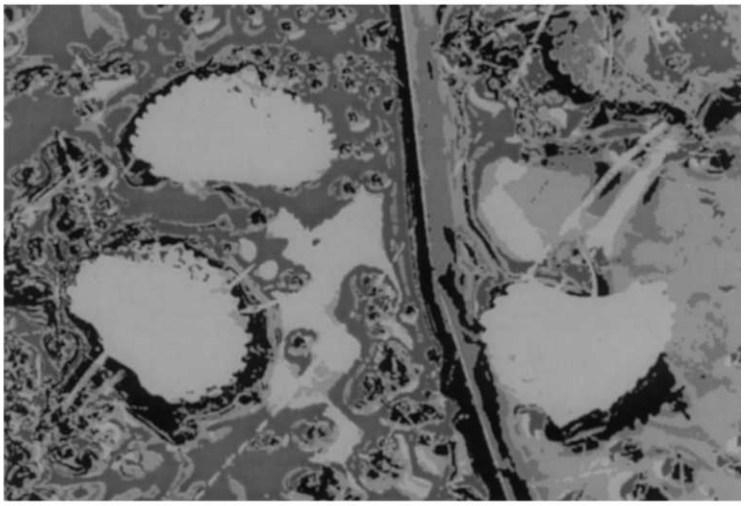
Fig. 10. Unsupervised segmentation of a 512×512 real image with two classes (red and green peppers). We can see some misclassification in the highlights (they were classified as green due to some greenish color in highlights). On the other hand, red reflections on green peppers are classified as red.

misclassification versus 9.19%). Considering supervised and unsupervised color segmentation results, we can see that the supervised method performs slightly better (0.05% misclassification versus 0.19%), especially when SNR increases. However, up to a certain SNR, the unsupervised results are very close to the supervised ones. For this particular image, this limit seems to be around SNR = 5 dB. In Table 1 we can compare supervised and unsupervised Gaussian parameters. One can see that final estimates are close to the supervised ones. The chief advantage of the unsupervised method is that it *does not* require human intervention. The findings were the same for real images.

In Table 2, we give the computing times in terms of the time required to perform pre-segmentation, quantization and unsupervised MRF segmentation. The most time consuming task is the MRF segmentation. Half of the computing



Original image (*leaves*).



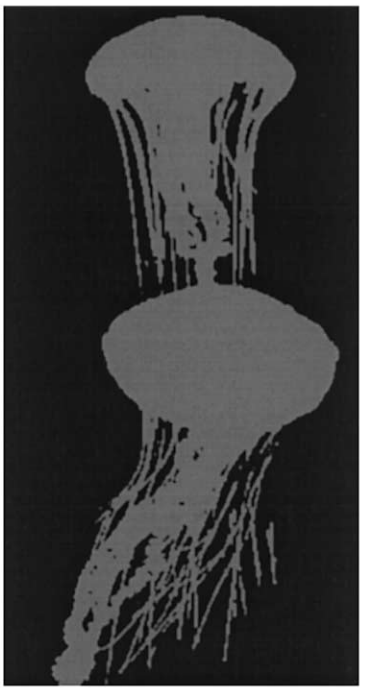
Unsupervised segmentation result (5 classes).

Fig. 11. Unsupervised segmentation of a 738 × 490 real image with five classes. One class corresponds to the blue water, another to the black shades and the remaining three classes to three different shades of yellow leaves.

power is used for the estimation of β. In average, 500 Metropolis iterations are executed until the the final estimate is obtained. The rest of the time is used by the estimation of Gaussian parameters and by the image segmentation step (Step (2) in our *Adaptive segmentation* algorithm. Obviously, larger images or images with more classes require more CPU time.

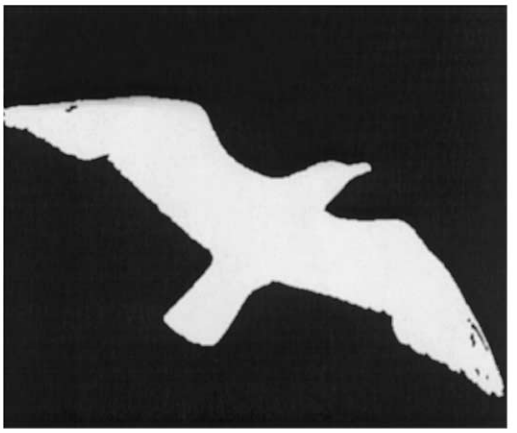
Finally, some results on real images are presented in Figs. 10 and 11. The algorithm also performs well in foreground/background segmentation (see Fig. 12).

Original image *jellies*.



Unsupervised segmentation result (2 classes).

Original image *seagull*.



Unsupervised segmentation result (2 classes).

Fig. 12. Unsupervised foreground/background segmentation of two real images. One can see that foreground objects have been quite well separated from the the background while fine details (especially on the *jellies* image) have been preserved.

6. Conclusion

In this paper, we have proposed an unsupervised color image segmentation algorithm. The segmentation model is defined in a Markovian framework and uses a first order potential derived from a tri-variate Gaussian distribution in order to tie the final segmentation to the observed image. To estimate the model-parameters, we use an iterative algorithm, which subsequently generates a labeling and then recomputes the parameter values. This process requires good initial mean values. Due to the large number of possible colors, the histogram cannot be analyzed directly, we have to re-quantize it without losing any important spatial information. To solve this problem, we have proposed a new method. It uses a pre-segmentation step based on color differences in order to reduce the number of colors. The method has been tested on a variety of real and synthetic images and the results are very close to supervised ones if SNR is kept reasonably high.

Acknowledgements

Some of the test images, used for evaluating the performance of the proposed algorithm, can be found at the Kodak Digital Image Offering WWW site at http://www.kodak.com/digitalImaging/samples/imageIntro.shtml.

References

Baxter, R.J., 1990. Exactly Solved Models in Statistical Mechinacis. Academic Press, New York.

Besag, J., 1986. On the statistical analysis of dirty pictures. J. Roy. Statist. Soc., Ser. B.

Daily, M.J., 1989. Color image segmentation using Markov random fields. In: Proc. DARPA Image Understanding.

Geman, D., 1985. Bayesian Image Analysis by Adaptive Annealing. In: Proc. IGARSS'85. Amherst, USA (October), pp. 269–277.

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. 6, 721–741.

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

Jain, A.K., 1989. Fundamentals of Digital Image Processing. Prentice Hall, Englewood Cliff, NJ.

Kato, Z., Berthod, M., Zerubia, J., 1996. A Hierarchical Markov Random Field Model and Multi-temperature Annealing for Parallel Image Classification. CVGIP Graphical Models Image Processing 58, 18–37.

Kato, Z., Zerubia, J., Berthod, M., 1999. Unsupervised parallel image classification using Markovian models. Pattern Recognition 32 (4), 591–604.

Lakshmanan, S., Derin, H., 1989. Simultaneous Parameter Estimation and Segmentation of Gibbs Random Fields Using Simulated Annealing. IEEE-PAMI 11, pp. 799–813.

Laarhoven, P.V., Aarts, E., 1987. Simulated annealing: theory and applications. Reidel Publication, Dordrecht, Holland.

Liu, J., Yang, Y.H., 1994. Multiresolution color image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 16, 689–700.

Panjwani, D.K., Healey, G., 1995. Markov random field models for unsupervised segmentation of textured color images. IEEE Trans. Pattern Anal. Mach. Intell. 17, 939–954.

Titterington, D.M., Smith, A.F.M., Makov, U.E., 1985. Statistical Analysis of Finite Mixture Distributions. Wiley, New York.

Verevka, O., Buchanan, J.W., 1995. Local *K*-means algorithm for color image quantization. In: Proc. GI'95, Quebec, Canada.