Markovian Image Models and their Application in Image Segmentation

Zoltan Kato

Department of Image Processing and Computer Graphics, Institute of Informatics, University of Szeged, Hungary

CONTENTS

Abstract

In this short thesis, we summarize the main results of our work related to Markovian image modeling:

1) We have proposed a novel hierarchical MRF model and applied it to graylevel image segmentation [1]–[6].
2) We have proposed a new annealing schedule for Simulated Annealing: Multi-temperature annealing allows to assign different temperatures to different cliques during the minimization of the energy of a MRF model. The convergence of the new algorithm has also been proved toward a global optimum [6], [7].
3) We have solved the estimation of the hierarchical model parameters and applied it to remote sensing image segmentation [8]–[11].
4) We have proposed a monogrid MRF model which is able to combine color and texture features in order to improve the quality of unsupervised segmentations [12], [13].
5) We have proposed a novel RJMCMC sampling method which is able to identify multi-dimensional Gaussian mixtures. This technique has been applied to fully automatic color image segmentation [14], [15].
6) A new multilayer MRF model has been proposed which is able to segment an image based on multiple cues (such as color, texture, or motion) [16]–[21].

Our other contributions not discussed in this thesis are as follows:

- We have developed a monogrid MRF model for graylevel image segmentation and applied the model for remote sensing imaging. For this model, we have proposed a novel pseudo-stochastic relaxation algorithm, called Modified Metropolis Dynamics (MMD) [22]–[24].
- We have studied different optimization techniques in the context of MRF energy minimization [22], [23], [25], [26]. As a result, we also developed a demo program implementing most of these algorithms. The program is freely downloadable from our website [27].
- In collaboration with SZTAKI, our monogrid MRF model and MMD algorithm [22]–[24] has been successfully adopted for the CNN (Cellular Neural Network) architecture [28], [29].

- We have developed a monogrid MRF model for color image segmentation [30] as well as for motion-based video object segmentation [31].
- We have successfully applied a non-photorealistic rendering technique, called Stochastic Paintbrush Transformation, to the problem of content based image retrieval [32], [33].

I. INTRODUCTION

THE primary goal of any segmentation algorithm is to divide the domain R of the input image into the disjoint parts R_i such that they belong to distinct objects in the scene. The solution of this problem sometimes requires high level knowledge about the shape and appearance of the objects under investigation [34]–[37]. In many applications, however, such information is not available or impractical to use. Hence low-level features of the surface patches are used for the segmentation process [38]–[40]. Herein, we are interested in the latter approach. In either case, we have to summarize all relevant information in a model which is then adjusted to fit the image data.

One broadly used class of models is the so called *cartoon model*, which has been extensively studied from both probabilistic [41] and variational [42], [43] 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* ω 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 Γ - the contour. The pair (ω, Γ) specifies a *segmentation*. Region based methods are mainly focusing on ω while edge based methods are trying to determine Γ directly.

Taking the probabilistic approach, one usually wants to come up with a *probability measure* on the set Ω of all possible segmentations of $\mathcal{I}$ and then select the one with the highest probability. Note that Ω is finite, although huge. A widely accepted standard, also motivated by the human visual system [44], [45], is to construct this probability measure in a Bayesian framework [46]–[48]: We shall assume that we have a set of observed (Y) and hidden (X) random variables. In our context, any observed value $y \in Y$ represents the low-level features used for partitioning the image, and the hidden entity $x \in X$ represents the segmentation itself. First, we have to quantify how well any occurrence of x fits y. This is expressed by the probability distribution $P(y|x)$ - the *imaging model*. Second, we define a set of properties that any segmentation x must posses regardless the image data. These are described by $P(x)$, the *prior*, which tells us how well any occurrence x satisfies these properties. Factoring these distributions and applying the Bayes theorem gives us the *posterior* distribution $P(x|y) \propto P(y|x)P(x)$. Note that the constant factor $1/P(y)$ has been dropped as we are only interested in $\widehat{x}$ 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 Θ. 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 [41], [49]. This is known in statistics as the *complete data* problem which is relatively easy to solve using Maximum Likelihood (ML) [46]. Although the prior knowledge of the parameters is a strong assumption, supervised methods are still useful alternatives when working in a controlled environment. Many industrial applications, like quality inspection of agricultural products [50], fall into this category. In the unsupervised case, however, we know neither Θ nor X. This is called the *incomplete data* problem where both Θ and X has to be inferred from the only observable entity Y. Hence our MAP estimation problem becomes $(\widehat{x}, \widehat{\Theta}) = \arg\max_{x,\Theta} P(x, \Theta|y)$. *Expectation Maximization* (EM) [51] and its variants (Stochastic EM [52], [53], Gibbsian EM [54]), as well as *Iterated Conditional Expectation* (ICE) [9], [55] are widely used to solve such problems. It is important to note, however, that these methods calculate a local maximum [46].

Due to the difficulty of estimating the number of pixel classes (or clusters), unsupervised algorithms often suppose that this parameter is *known a priori* [40], [53], [56]–[58]. 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.

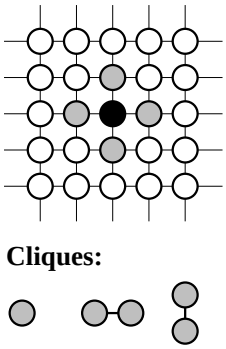


Fig. 1. First order neighborhood system with corresponding cliques.

A. Markovian Approach

In real images regions are usually homogeneous, neighboring pixels have similar properties. Markov Random Fields (MRF) are often used to capture such contextual constraints in a probabilistic framework. MRFs are well studied with a strong theoretical background hence providing a tool for rigorous and concise image modeling. Furthermore, they allow Markov Chain Monte Carlo (MCMC) sampling of the (hidden) underlying structure which greatly simplifies inference and parameter estimation.

Formally, a simple MRF image model is constructed as follows: we are given a set of sites (usually corresponding to pixels) $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$. For each site s, the region-type (or class) that the site belongs to is specified by a class label, ω_s, which is modeled as a discrete random variable taking values in $\Lambda = \{1, 2, \ldots, L\}$. 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 (*e.g.* graylevel, color, texture,...) are supposed to be a realization $\mathcal{F} = \{f_s | s \in \mathcal{S}\}$ from another random field, which is a function of the label process ω. 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 ω and the observable noisy image process $\mathcal{F}$. If each pixel class is represented by a different model then the observed image may be viewed as a sample from a realization of the underlying label field.

$(\omega, \mathcal{F})$ is then regarded as a MRF with respect to an appropriate neighborhood-system $\mathcal{G} = \{\mathcal{G}_s\}_{s \in \mathcal{S}}$. The simplest example of such a neighborhood can be seen in Fig. 1. According to the Hammersley-Clifford theorem [59], $(\omega, \mathcal{F})$ must then follow a Gibbs distribution with an energy function $U(\omega, \mathcal{F}) = \sum_{C \in \mathcal{C}} V_C(\omega, \mathcal{F})$, where C denotes a clique of $\mathcal{G}$, and $\mathcal{C}$ is the set of all cliques. The restriction of ω to the sites of a given clique C is denoted by ω_C. The potential function $V_C(\omega_C)$ is defined for every $C \in \mathcal{C}$ and every $\omega \in \Omega$, where $\Omega = \Lambda^N$ is the set of all possible L^N discrete labelings. The advantage of such a decomposition is that these potentials are a function of the local configuration of the field making it possible to define the Gibbs distribution directly in terms of local interactions.

The MAP estimate $\hat{\omega}$ of the label field is then obtained by minimizing the non-convex energy function, which can be solved by stochastic or deterministic relaxation [22], [23], [25], [26].

II. Hierarchical MRF Models and Multi-Temperature Annealing

It is well known that multigrid methods can improve significantly the convergence rate and the quality of the final results of iterative relaxation techniques. Herein, we propose a new hierarchical model [1]–[6], which consists of a label pyramid and a single observation field. The parameters of the coarse grid can be derived by simple computation from the finest grid. In addition, we have introduced a new local interaction between two neighboring grids which allows to propagate information more efficiently giving estimates closer to the global optimum for deterministic as well as for stochastic relaxation schemes. For the hierarchical model, we also propose a novel Multi-Temperature Annealing (MTA) algorithm [6]–[8].

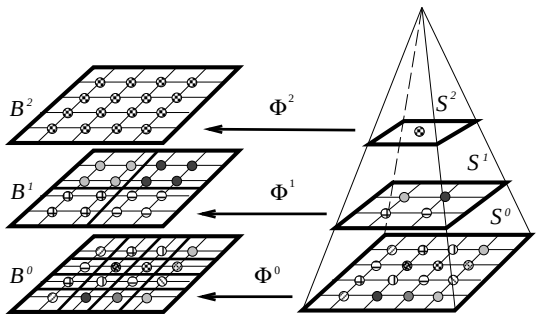


Fig. 2. The isomorphism Φ^i between $\mathcal{B}^i$ and $\mathcal{S}^i$.

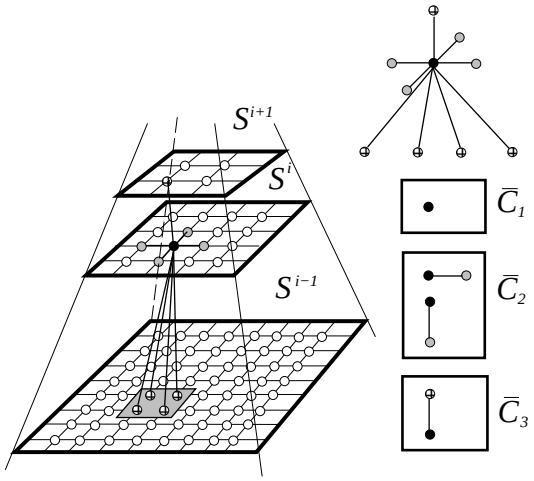


Fig. 3. The neighborhood system $\bar{\mathcal{G}}$ and the cliques $\bar{\mathcal{C}}_1$, $\bar{\mathcal{C}}_2$ and $\bar{\mathcal{C}}_3$.

The convergence towards the global optimum is proven by the generalization of the annealing theorem of Geman and Geman [41].

A. Multiscale and Hierarchical Model

In the following, we will focus on a MRF with a first order neighborhood (see Fig. 1) whose energy function is given by:

$$U(\omega, \mathcal{F}) = U_1(\omega, \mathcal{F}) + U_2(\omega) \tag{1}$$

where U_1 (resp. U_2) denotes the energy of the first order (resp. second order) cliques. To generate a multigrid MRF model, let us divide the initial grid into blocks of $n \times n$, typically 16 (4×4) neighboring pixels. We consider that the same label is assigned to each pixels of a given block. These configurations will describe the MRF at scale 1. Scale i is defined similarly by considering labels which are constant over blocks of size $n^i \times n^i$.

Let $\mathcal{B}^i = \{b_1^i, \ldots, b_{N_i}^i\}$ ($N_i = N/n^{2i}$) denote the set of blocks and Ω_i the configuration-space at scale i ($\Omega_i \subset \Omega_{i-1} \subset \cdots \subset \Omega_0 = \Omega$). The label associated with block b_k^i is denoted by ω_k^i. We can define the same neighborhood structure on $\mathcal{B}^i$ as on $\mathcal{S}$:

$$b_k^i \text{ and } b_l^i \text{ are neighbors} \Longleftrightarrow \left\{ \begin{array}{l} b_k^i \equiv b_l^i \text{ or} \\ \exists C \in \mathcal{C}\text{: } C \cap b_k^i \neq \emptyset \text{ and } C \cap b_l^i \neq \emptyset \end{array} \right. \tag{2}$$

Let us partition the original set $\mathcal{C}$ into two disjoint subsets $\mathcal{C}_k^i$ (cliques which are included in b_k^i) and $\mathcal{C}_{k,l}^i$(cliques which sit astride two neighboring blocks $\{b_k^i, b_l^i\}$). It is obvious from this partition that our energy function can be decomposed in the following way:

$$U_1(\omega, \mathcal{F}) = \sum_{s \in \mathcal{S}} V_1(\omega_s, f_s) = \sum_{b_k^i \in \mathcal{B}^i} \underbrace{\sum_{s \in b_k^i} V_1(\omega_s, f_s)}_{V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F})} = \sum_{b_k^i \in \mathcal{B}^i} V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F}) \tag{3}$$

$$\begin{aligned} U_2(\omega) = \sum_{C \in \mathcal{C}} V_2(\omega_c) &= \sum_{b_k^i \in \mathcal{B}^i} \underbrace{\sum_{C \in \mathcal{C}_k^i} V_2(\omega_c)}_{V_k^{\mathcal{B}^i}(\omega_k^i)} + \sum_{\{b_k, b_l\} neighbors} \underbrace{\sum_{C \in \mathcal{C}_{k,l}^i} V_2(\omega_c)}_{V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i)} \\ &= \sum_{b_k^i \in \mathcal{B}^i} V_k^{\mathcal{B}^i}(\omega_k^i) + \sum_{\{b_k, b_l\} neighbors} V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i) \end{aligned} \tag{4}$$

Now, we define a pyramid (see Figure 2) where level i contains the coarse grid $\mathcal{S}^i$ which is isomorphic to the scale $\mathcal{B}^i$. The coarse grid has a reduced configuration space $\Xi^i = \Lambda^{N_i}$. The isomorphism $\Phi^i : \mathcal{S}^i \rightarrow \mathcal{B}^i$ is just a projection of the coarse label field to the finest grid $\mathcal{S}^0 \equiv \mathcal{S}$. The energy function on the grid $\mathcal{S}^i$ $(i = 0, \ldots, M)$ is derived from Eq. (3)–(4):

$$\begin{aligned} U^i(\omega^i, \mathcal{F}) &= U_1^i(\omega^i, \mathcal{F}) + U_2^i(\omega^i) = U_1(\Phi^i(\omega_i), \mathcal{F}) + U_2(\Phi^i(\omega_i)) \\ \text{where } U_1^i(\omega^i, \mathcal{F}) &= \sum_{k \in \mathcal{S}^i} (V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F}) + V_k^{\mathcal{B}^i}(\omega_k^i)) = \sum_{k \in \mathcal{S}^i} V_1^i(\omega_k^i, \mathcal{F}) \quad (5) \\ \text{and } U_2^i(\omega^i) &= \sum_{\{k,l\} neighbors} V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i) = \sum_{C^i \in \mathcal{C}^i} V_2^i(\omega_C^i) \quad (6) \end{aligned}$$

where C^i is a second order clique corresponding to the definition in Eq. (2) and $\mathcal{C}^i$ is the set of cliques on grid i.

Let $\bar{\mathcal{S}} = \{\bar{s}_1, \ldots, \bar{s}_{\bar{N}}\} = \bigcup_{i=0}^{M} \mathcal{S}^i$ ($\bar{N} = \sum_{i=0}^{M} N^i$) denote the sites of the pyramid. We define the following function Ψ between two neighboring levels, which assigns to a site its descendants (that is the sites of the corresponding block):

$$\Psi : \mathcal{S}^i \longrightarrow \mathcal{S}^{i-1}, \quad \Psi(\bar{s}) = \{\bar{r} \mid \bar{s} \in \mathcal{S}^i \Rightarrow \bar{r} \in \mathcal{S}^{i-1} \text{ and } b_{\bar{r}}^{i-1} \subset b_{\bar{s}}^i\} \quad (7)$$

It is clear that Ψ^{-1} will assign to a site its ancestor (that is the site at the upper level corresponding to the block of this site). Now we can define on these sites the following neighborhood-system (see Fig. 3):

$$\bar{\mathcal{G}} = (\bigcup_{i=0}^{M} \mathcal{G}^i) \cup \{\Psi^{-1}(\bar{s}) \cup \Psi(\bar{s}) \mid \bar{s} \in \bar{\mathcal{S}}\} \quad (8)$$

where $\mathcal{G}^i$ is the neighborhood structure of the i^{th} level. We will consider only the first and second order cliques, potentials for other cliques are supposed to be 0. Let $\bar{\mathcal{C}}$ denote the set of these cliques which can be partitioned into three disjoint subsets $\bar{\mathcal{C}}_1, \bar{\mathcal{C}}_2, \bar{\mathcal{C}}_3$ corresponding to first order cliques, second order cliques which are on the same level and second order cliques which sit astride two neighboring levels (see Figure 3). Let $\bar{\Omega}$ denote the configuration-space of the pyramid:

$$\bar{\Omega} = \Xi^0 \times \Xi^1 \times \cdots \times \Xi^M = \{\bar{\omega} \mid \bar{\omega} = (\omega^0, \omega^1, \ldots, \omega^M)\} \quad (9)$$

The model on the pyramid defines a MRF, whose energy function is given by:

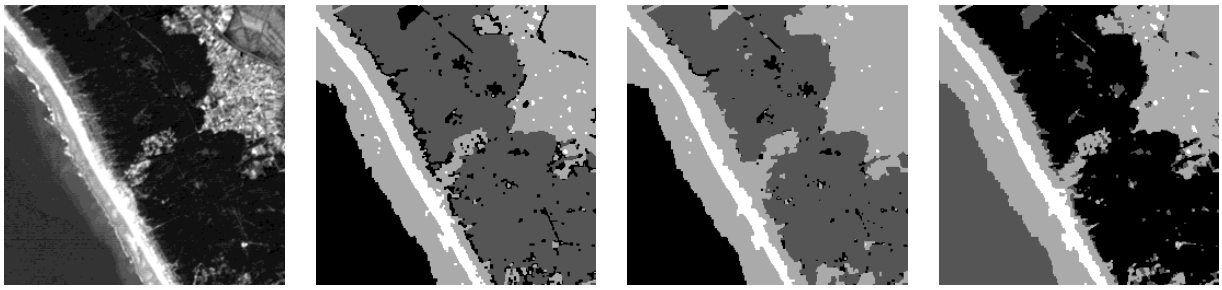
$$\begin{aligned} \bar{U}(\bar{\omega}, \mathcal{F}) &= \bar{U}_1(\bar{\omega}, \mathcal{F}) + \bar{U}_2(\bar{\omega}) \quad (10) \\ \bar{U}_1(\bar{\omega}, \mathcal{F}) &= \sum_{\bar{s} \in \bar{\mathcal{S}}} \bar{V}_1(\bar{\omega}_{\bar{s}}, \mathcal{F}) = \sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i} V_1^i(\omega_{s^i}^i, \mathcal{F}) = \sum_{i=0}^{M} U_1^i(\omega^i, \mathcal{F}) \\ \bar{U}_2(\bar{\omega}) &= \sum_{C \in \bar{\mathcal{C}}_2} \bar{V}_2(\bar{\omega}_C) + \sum_{C \in \bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C) = \sum_{i=0}^{M} U_2^i(\omega^i) + \sum_{C \in \bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C) = \sum_{i=0}^{M} \sum_{C \in \mathcal{C}^i} V_2^i(\omega_c^i) + \sum_{C \in \bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C) \end{aligned}$$

The above energy of the hierarchical model can be minimized using classical combinatorial optimization algorithms [22], [23], [25], [26], [61]. The only difference is that we work on a pyramid here and not on a rectangular lattice as in the case of classical monogrid models. We have applied the model for supervised image segmentation and compared the segmentation results of the classical monogrid [22]–[24], [26], multiscale and hierarchical models on synthetic (Fig. 4) and real (Fig. 5) images. For both images, the label pyramid has been generated with 4 levels. The detailed equations can be found in [1], [6]. All tests have been conducted on a Connection Machine CM200 with $8K$ processors. In terms of segmentation quality, the hierarchical model clearly outperforms the other methods. Further results can be found in [1], [6].

model	num. of iter.	CPU time	time/iter.	error rate	β	γ
monogrid	89	10.39 sec.	0.117 sec.	2576	1.0	—
multiscale	146	14.7 sec.	0.1 sec.	2118	1.0	—
hierarchical	42	460.9 sec.	10.97 sec.	1231	1.0	0.2

Noisy image ($SNR = 10dB$) Monogrid Multiscale Hierarchical

Fig. 4. Results obtained by the Gibbs Sampler [41] on a noisy synthetic image (128×128, $SNR = 10dB$) with 16 classes [1]–[6]. In the table, we show for each model the number of iterations, the CPU time, the error rate of the segmentation (= the number of misclassified pixels) and the inter- and intra-clique potentials β and γ.



Original image Monogrid Multiscale Hierarchical

Fig. 5. Results obtained by ICM [60] on a (256×256) SPOT image with 4 classes [1]–[6].

B. Multi-Temperature Annealing

In the following we will focus on Simulated Annealing (SA) [41], where the temperature-change is controlled by the so-called *annealing schedule*. There are two well known schemes, *homogeneous* and *inhomogeneous* annealing [61], which works also on the hierarchical model. Herein, we propose a new annealing schedule, called *Multi-Temperature Annealing* (MTA), which is the most efficient with the new model. The basic idea is to associate higher temperatures to coarser levels in the pyramid which makes the algorithm less sensitive to local minima. However at a finer resolution, the relaxation is performed at a lower temperature (at the bottom level, it is close to 0). For the cliques siting between two levels, we use either the temperature of the finer level or the one of the coarser level (but once chosen, we always keep the same choice throughout the algorithm). More generally, we have the following problem:

Let $\mathcal{S} = \{s_1, \ldots, s_N\}$ be a set of sites, $\mathcal{G}$ some neighborhood system with cliques $\mathcal{C}$ and ω a MRF over these sites with energy function U. π_0 denotes the uniform distribution on the set of globally optimal configurations, and define $U^{sup} = \max_\omega U(\omega)$, $U^{inf} = \min_\omega U(\omega)$ and $\Delta = U^{sup} - U^{inf}$. Furthermore, let us suppose that the sites are visited for updating in the order $\{n_1, n_2, \ldots\} \subset \mathcal{S}$. We now define an annealing scheme where the temperature T depends on the iteration k and on the cliques C. For that purpose, let $\oslash$ denotes the following operation:

$$P(X = \omega) = \pi_{T(k,C)}(\omega) \quad = \quad \frac{\exp(-U(\omega) \oslash T(k, C))}{Z} \tag{11}$$

$$\text{where } U(\omega) \oslash T(k,C) \;=\; \sum_{C\in\mathcal{C}} \frac{V_C(\omega)}{T(k,C)}\,. \tag{12}$$

As usual with SA [41], [61], the transition from one configuration to another is governed by the energy change between the two states. Assuming $\omega' \in \Omega_{opt}$ is a *globally* optimal configuration, $U(\omega') - U^{inf}$ equals to 0 (*i.e.* there is no more energy change, the system is frozen). In the case of a classical annealing, dividing by a constant temperature does not change this relation (obviously, $\forall k\colon (U(\omega') - U^{inf})/T_k$ is still 0). But it is not necessarily true that $(U(\omega') - U^{inf}) \oslash T(k,C)$ is also 0! Because choosing sufficiently small temperatures for the cliques where ω'_C is *locally* not optimal (*i.e.* strengthening the non-optimal cliques) and choosing sufficiently high temperatures for the cliques where ω'_C is *locally* optimal (*i.e.* weakening the optimal cliques), we obtain $(U(\omega') - U^{inf}) \oslash T(k,C) > 0$, meaning that ω' is no longer *globally* optimal (*i.e.* in such cases, SA may not be able to reach a global optimum).

Thus, we have to impose further conditions on the temperature to guarantee the convergence toward global optimum. First, let us examine the decomposition over the cliques of $U(\omega) - U(\eta)$ for arbitrary ω and η, $\omega \neq \eta$:

$$U(\omega) - U(\eta) = \sum_{C\in\mathcal{C}} (V_C(\omega) - V_C(\eta)). \tag{13}$$

Indeed, there may be negative and positive members in the decomposition. According to this fact, we have the following subsums:

$$\sum_{C\in\mathcal{C}} (V_C(\omega) - V_C(\eta)) = \underbrace{\sum_{C\in\mathcal{C}:(V_C(\omega)-V_C(\eta))<0} (V_C(\omega) - V_C(\eta))}_{\Sigma^-(\omega,\eta)} + \underbrace{\sum_{C\in\mathcal{C}:(V_C(\omega)-V_C(\eta))\geq 0} (V_C(\omega) - V_C(\eta))}_{\Sigma^+(\omega,\eta)}\,. \tag{14}$$

Furthermore, let us define Σ^+_Δ as:

$$\Sigma^+_\Delta = \min_{\substack{\omega' \in \Omega_{sup} \\ \omega'' \in \Omega_{opt}}} \Sigma^+(\omega', \omega''). \tag{15}$$

Then the following theorem gives an annealing schedule, where *the temperature is a function of* k *and* $C \in \mathcal{C}$ [6]:

Theorem 1 (Multi-Temperature Annealing): Assume that there exists an integer $\kappa \geq N$ such that for every $k = 0, 1, 2, \ldots$, $\mathcal{S} \subseteq \{n_{k+1}, n_{k+2}, \ldots, n_{k+\kappa}\}$. For all $C \in \mathcal{C}$, let $T(k,C)$ be any decreasing sequence of temperatures in k for which

1) $\lim_{k\to\infty} T(k,C) = 0$.
   Let us denote respectively by T_k^{inf} and T_k^{sup} the maximum and minimum of the temperature function at k ($\forall C \in \mathcal{C}\colon T_k^{inf} \leq T(k,C) \leq T_k^{sup}$).
2) For all $k \geq k_0$, for some integer $k_0 \geq 2$: $T_k^{inf} \geq N\Sigma^+_\Delta / \ln(k)$.
3) If $\Sigma^-(\omega,\omega') \neq 0$ for some $\omega \in \Omega \setminus \Omega_{opt}$, $\omega' \in \Omega_{opt}$ then a further condition must be imposed:
   For all k: $\frac{T_k^{sup} - T_k^{inf}}{T_k^{inf}} \leq R$ with

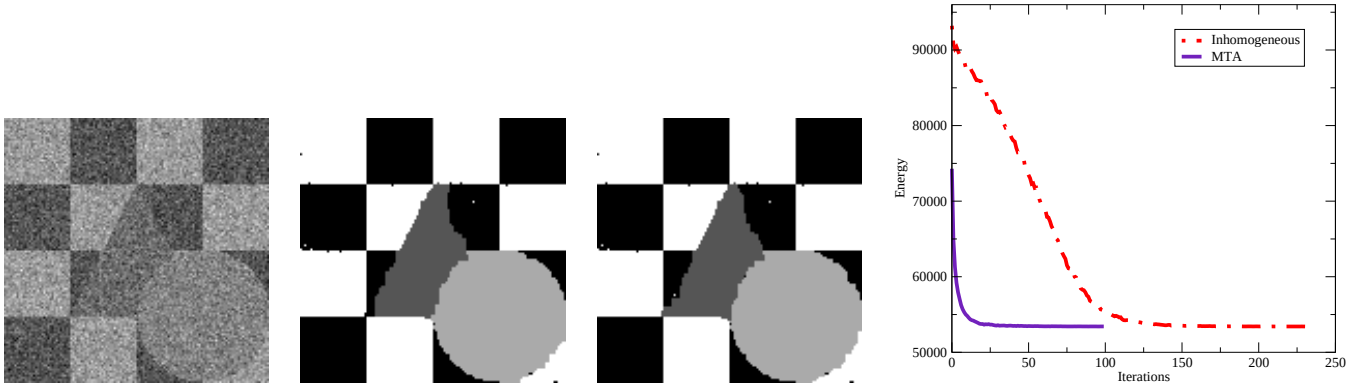
$$R = \min_{\substack{\omega \in \Omega \setminus \Omega_{opt} \\ \omega' \in \Omega_{opt} \\ \Sigma^-(\omega,\omega') \neq 0}} \frac{U(\omega) - U^{inf}}{\mid \Sigma^-(\omega,\omega') \mid}. \tag{16}$$

Then for any starting configuration $\eta \in \Omega$ and for every $\omega \in \Omega$:

$$\lim_{k\to\infty} P(X(k) = \omega \mid X(0) = \eta) = \pi_0(\omega). \tag{17}$$

The complete proof of this theorem can be found in [3], [6].

Remarks:



SNR= $3dB$ Inhomogeneous MTA Energy plots

Fig. 6. Energy decrease and segmentation results of the Gibbs sampler on a synthetic image with the inhomogeneous and MTA schedules. In both cases, the parameters were strictly the same, the only difference is the applied schedule. We also show the global energy plot (computed at a fixed temperature on the finest level) versus the number of iterations. Note that both schedules reach practically the same minimum (53415.4 for the inhomogeneous and 53421.4 for the MTA), however the inhomogeneous schedule requires 238 iterations (796.8 sec. CPU time) while the MTA schedule requires only 100 iterations (340.6 sec. CPU time) for the convergence [3], [6], [7].

1) In practice, we cannot determine R and Σ_{Δ}^{+}, as we cannot compute Δ neither.
2) Considering Σ_{Δ}^{+} in condition 2, we have the same problem as in the case of a classical annealing. The only difference is that in a classical annealing, we have Δ instead of Σ_{Δ}^{+}. Consequently, the same solutions may be used: an exponential schedule with a sufficiently high initial temperature.
3) The factor R is more interesting. We propose herein two possibilities which can be used for practical implementations of the method: Either we choose a sufficiently small interval $[T_0^{inf}, T_0^{sup}]$ and suppose that it satisfies the condition 3 (we have used this technique in the simulations), or we use a more strict but easily verifiable condition instead of condition 3, namely:

$$\lim_{k\to\infty} \frac{T_k^{sup} - T_k^{inf}}{T_k^{inf}} = 0. \tag{18}$$

4) What happens if $\Sigma^{-}(\omega, \omega')$ is zero for all ω and ω' in condition 3 and thus R is not defined? This is the best case because it means that all *globally* optimal configurations are also *locally* optimal. That is we have no restriction on the interval $[T_k^{inf}, T_k^{sup}]$, thus any *local* temperature schedule satisfying conditions 1–2 is good.

In Fig. 6, we compare the inhomogeneous and MTA schedules on a noisy synthetic image using the Gibbs sampler. Since each site interacts with its ancestor and its descendants, the hierarchical model usually requires more computing time than a monogrid model. However, as we have shown in [3], [6], [7], experiments prove that this model with the MTA schedule yields faster convergence (with respect to the number of iterations) for the stochastic relaxation algorithms and gives estimates which are closer to the global optimum. Other tests can be found in [3], [6], [7].

III. Parameter Estimation

In real life applications, the model parameters are usually unknown, one has to estimate [62] them from the observable image. Here we develop an algorithm for hierarchical Markovian models [8]–[11]. Our approach is similar in spirit to Iterative Conditional Estimation [53], [63] as well as to the Estimation-Maximization algorithm: we recursively look at the Maximum a Posteriori (MAP) estimate of the label field given the estimated parameters then we look at the Maximum Likelihood (ML) estimate of the parameters given a tentative labeling obtained in the previous step. The only parameter supposed to be known is the number of labels, all the other parameters are estimated.

When both the model parameters Θ and ω are unknown, the estimation problem becomes [11], [40], [64]

$$(\widehat{\omega}, \widehat{\Theta}) = \arg\max_{\omega,\Theta} P(\omega, \mathcal{F} \mid \Theta). \tag{19}$$

The pair $(\widehat{\omega}, \widehat{\Theta})$ is the global maximum of the joint probability $P(\omega, \mathcal{F} \mid \Theta)$. If we regard Θ as a random variable, the above maximization is an ordinary MAP estimation in the following way [64]: Let us suppose, that Θ is restricted to a finite volume domain $\mathcal{D}_\Theta$ and suppose that Θ is uniform on $\mathcal{D}_\Theta$ (that is $P(\Theta)$ is constant). Then, we get [11], [64]:

$$\arg\max_{\omega,\Theta} P(\omega, \Theta \mid \mathcal{F}) = \arg\max_{\omega,\Theta} \frac{P(\omega, \mathcal{F} \mid \Theta)P(\Theta)}{P(\mathcal{F})} = \arg\max_{\omega,\Theta} \frac{P(\omega, \mathcal{F} \mid \Theta)}{\int_{\mathcal{D}_\Theta} \sum_{\omega\in\Omega} P(\omega, \mathcal{F} \mid \Theta) d\Theta} \tag{20}$$

$$= \arg\max_{\omega,\Theta} P(\omega, \mathcal{F} \mid \Theta). \tag{21}$$

However, this maximization is very difficult, having no direct solution. Even Simulated Annealing (SA) is not implementable because the local characteristics with respect to the parameters Θ cannot be computed from $P(\omega, \mathcal{F} \mid \Theta)$. One possible solution is to adopt the following criterion instead [11], [40], [64]:

$$\widehat{\omega} = \arg\max_{\omega} P(\omega, \mathcal{F} \mid \widehat{\Theta}) \tag{22}$$

$$\widehat{\Theta} = \arg\max_{\Theta} P(\widehat{\omega}, \mathcal{F} \mid \Theta) \tag{23}$$

Clearly, Eq. (22) is equivalent to Eq. (19) for $\Theta = \widehat{\Theta}$ and Eq. (23) is equivalent to Eq. (19) with $\omega = \widehat{\omega}$. Furthermore, Eq. (22) is equivalent to the MAP estimate of ω in the case of known parameters:

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

Hence in the following we will concentrate on Eq. (23) which gives the ML estimate of the parameters. Considering the hierarchical MRF segmentation model (see Fig. 3), we have the following logarithmic likelihood function [8]–[11]:

$$\sum_{i=0}^{M} \sum_{s^i\in\mathcal{S}^i} \sum_{s\in b^i_{s^i}} \left(-\ln(\sqrt{2\pi}\sigma_{\widehat{\omega}_s}) - \frac{(f_s - \mu_{\widehat{\omega}_s})^2}{2\sigma^2_{\widehat{\omega}_s}} \right) - \beta \underbrace{\sum_{i=0}^{M} q^i \sum_{C^i\in\mathcal{C}^i} \delta(\widehat{\omega}_{C^i})}_{N^{ih}(\widehat{\omega})} - \gamma \underbrace{\sum_{C\in\bar{\mathcal{C}}_3} \delta(\widehat{\omega}_C)}_{\bar{N}^{ih}(\widehat{\omega})} - \ln(Z(\beta,\gamma)) \tag{24}$$

where q^i is the number of cliques between two neighboring blocks at scale $\mathcal{B}^i$, $N^{ih}(\widehat{\omega})$ denotes the number of inhomogeneous cliques siting at the same scale and $\bar{N}^{ih}(\widehat{\omega})$ denotes the number of inhomogeneous cliques siting astride two neighboring levels in the pyramid. Considering the first term, we get

$$\sum_{i=0}^{M} \sum_{s^i\in\mathcal{S}^i} \sum_{s\in b^i_{s^i}} \left(-\ln(\sqrt{2\pi}\sigma_{\widehat{\omega}_s}) - \frac{(f_s - \mu_{\widehat{\omega}_s})^2}{2\sigma^2_{\widehat{\omega}_s}} \right) = \sum_{\lambda\in\Lambda} \sum_{i=0}^{M} \sum_{s^i\in\mathcal{S}^i_\lambda} \sum_{s\in b^i_{s^i}} \left(-\ln(\sqrt{2\pi}\sigma_\lambda) - \frac{(f_s - \mu_\lambda)^2}{2\sigma^2_\lambda} \right) \tag{25}$$

where $\mathcal{S}^i_\lambda$ is the set of sites at level i where $\widehat{\omega}_{s^i} = \lambda$. Derivating with respect to μ_λ and σ_λ, we get a closed form solution for the ML estimates of the Gaussian parameters:

$$\forall\lambda\in\Lambda: \quad \mu_\lambda = \frac{1}{\sum_{i=0}^{M} \mid \mathcal{S}^i_\lambda \mid} \sum_{i=0}^{M} \sum_{s^i\in\mathcal{S}^i_\lambda} \sum_{s\in b^i_{s^i}} f_s, \quad \sigma^2_\lambda = \frac{1}{\sum_{i=0}^{M} \mid \mathcal{S}^i_\lambda \mid} \sum_{i=0}^{M} \sum_{s^i\in\mathcal{S}^i_\lambda} \sum_{s\in b^i_{s^i}} (f_s - \mu_\lambda)^2 \tag{26}$$

Notice that a grey-level value f_s may be considered several times. More precisely, f_s is considered m-times in the above sum for a given λ if there are m scales where $\widehat{\omega}$ assigns the label λ to the site s. m can also be seen as a weight. Obviously, the more s has been labeled by λ at different levels, the more is

probable that s belongs to class λ and hence its grey-level value f_s characterizes better the class λ. The derivates of the logarithmic likelihood function with respect to β and γ are given by:

$$\frac{\partial}{\partial\beta}\left(-\beta N^{ih}(\widehat{\omega}) - \ln(Z(\beta,\gamma))\right) = -N^{ih}(\widehat{\omega}) - \frac{\partial}{\partial\beta}\ln(Z(\beta,\gamma)) \quad (27)$$

$$\frac{\partial}{\partial\gamma}\left(-\gamma \bar{N}^{ih}(\widehat{\omega}) - \ln(Z(\beta,\gamma))\right) = -\bar{N}^{ih}(\widehat{\omega}) - \frac{\partial}{\partial\gamma}\ln(Z(\beta,\gamma)) \quad (28)$$

From which, we get

$$N^{ih}(\widehat{\omega}) = \frac{\sum_{\omega\in\Omega} N^{ih}(\omega)\exp(-\beta N^{ih}(\omega) - \gamma\bar{N}^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega) - \gamma\bar{N}^{ih}(\omega))} \quad (29)$$

$$\bar{N}^{ih}(\widehat{\omega}) = \frac{\sum_{\omega\in\Omega} \bar{N}^{ih}(\omega)\exp(-\beta N^{ih}(\omega) - \gamma\bar{N}^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega) - \gamma\bar{N}^{ih}(\omega))} \quad (30)$$

The solution of the above equations can be approximated using the following algorithm.

Algorithm 1 (Hyperparameter Estimation):

① Set $k = 0$ and initialize $\widehat{\beta}^0$ and $\widehat{\gamma}^0$. Furthermore, let $N^{ih}(\widehat{\omega})$ denote the number of inhomogeneous cliques at the same scale and $\bar{N}^{ih}(\widehat{\omega})$ denotes the number of inhomogeneous cliques between levels.

② Using SA at a fixed temperature T, generate a new labeling η sampling from

$$P(\mathcal{X}=\omega) = \frac{\exp\left(-\frac{\widehat{\beta}^k}{T}\sum_{i=0}^{M}\sum_{\{s,r\}\in\mathcal{C}^i}\delta(\omega_s,\omega_r)\right)}{Z(\widehat{\beta}^k,\widehat{\gamma}^k)} + \frac{\exp\left(-\frac{\widehat{\gamma}^k}{T}\sum_{\{s,r\}\in\bar{\mathcal{C}}}\delta(\omega_s,\omega_r)\right)}{Z(\widehat{\beta}^k,\widehat{\gamma}^k)}. \quad (31)$$

Compute the number of inhomogeneous cliques $N^{ih}(\eta)$ and $\bar{N}^{ih}(\eta)$ in η.

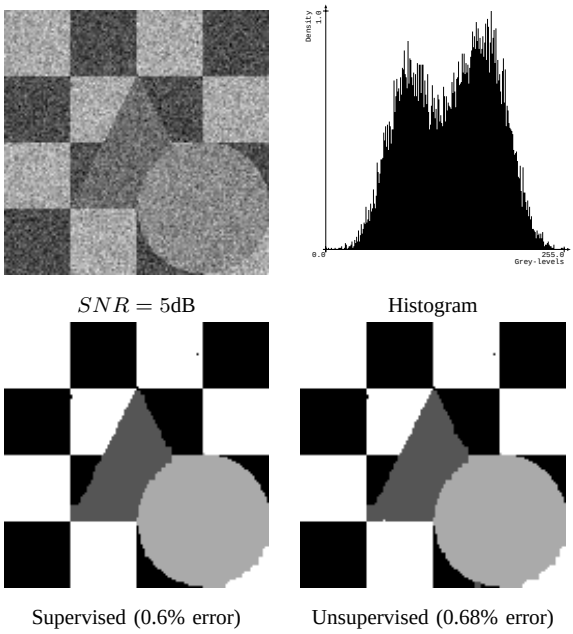
③ If $N^{ih}(\eta) \approx N^{ih}(\widehat{\omega})$ *and* $\bar{N}^{ih}(\eta) \approx \bar{N}^{ih}(\widehat{\omega})$ then stop, else $k = k+1$. If $N^{ih}(\eta) < N^{ih}(\widehat{\omega})$ then decrease $\widehat{\beta}^k$, if $N^{ih}(\eta) > N^{ih}(\widehat{\omega})$ then increase $\widehat{\beta}^k$. $\widehat{\gamma}^k$ is obtained in the same way. Continue Step ② with $(\widehat{\beta}^k,\widehat{\gamma}^k)$.

This algorithm completes the computation of the ML estimate of the parameters given $\widehat{\omega}$. The unsupervised segmentation is then carried out using *Adaptive Simulated Annealing* [11], [64], which is an iterative algorithm generating tentative labelings based on current parameter estimates (*i.e.* solving Eq. (22)) then updating the parameter values to their ML estimate based on the current labeling (*i.e.* solving Eq. (23) by making use of Eq. (26) and Algorithm 1). In fact, it is a classical Simulated Annealing with an additional step to reestimate model parameters during segmentation. The convergence of ASA has been proven in [40].

The algorithm has been tested on several synthetic and real images [9]–[11]. In Fig. 7, we show one of these results. In summary, the presented unsupervised algorithm provide results comparable to those obtained by supervised segmentations, but of course at the price of higher computing time.

IV. Unsupervised Segmentation of Color Textured Images

The proposed segmentation model [12], [13] consists of a monogrid MRF defined over a nearest neighborhood system (see Fig. 1) with pixel classes represented by multivariate Gaussian distributions. This kind of modelization corresponds well to our features: Texture feature images (extracted by Gabor filters) 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, pixels with similar color map to their average color. 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. Applying these ideas, the *image process* $\mathcal{F}$ can be formalized as follows: $P(\vec{\boldsymbol{f}}_s \mid \omega_s)$ follows a Normal distribution $N(\vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma})$, each pixel class $\lambda \in \Lambda = \{1, 2, \ldots, L\}$ is represented by its mean vector $\vec{\boldsymbol{\mu}}_\lambda$ and covariance matrix $\boldsymbol{\Sigma}_\lambda$. The whole posterior



Parameter	Unsupervised		Supervised
	Initial	Final	
μ_0	83.5	84.3	85.48
σ_0^2	256.0	483.9	446.60
μ_1	100.0	115.5	115.60
σ_1^2	169.0	444.6	533.97
μ_2	152.5	146.7	146.11
σ_2^2	676.0	502.1	540.32
μ_3	181.5	177.9	178.01
σ_3^2	100.0	500.0	504.34
β	0.7	1.0	0.7
γ	0.1	0.1	0.1

Fig. 7. Supervised and unsupervised segmentation results and misclassification rate with the Gibbs Sampler. We also compare the parameters obtained by the unsupervised algorithm to the ones used for the supervised segmentation [8]–[11].

can now 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. Thus the energy function of the so defined MRF image segmentation model has the following form:

$$\sum_{s\in\mathcal{S}} \left(\ln(\sqrt{(2\pi)^n \mid \boldsymbol{\Sigma}_{\omega_s} \mid}) + \frac{1}{2}(\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s})\boldsymbol{\Sigma}_{\omega_s}^{-1}(\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s})^T \right) + \beta \sum_{\{s,r\}\in\mathcal{C}} \delta(\omega_s, \omega_r) \tag{32}$$

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

The proposed segmentation model has the following parameters:

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

The automatic determination of L will be addressed in Section V. While L strongly depends on the input image data, β is largely independent of it. Experimental evidence suggests that the model is not sensitive to a particular setting of β [12], [13]. 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. Our solution to this problem [13] adopts a general iterative algorithm, known as the *EM algorithm*, to compute the maximum likelihood estimates of the parameters of a mixture density. Basically, we will fit a Gaussian mixture of L components to the histogram of the image features. The observations consist of the histogram data $\vec{\boldsymbol{d}}_i (i = 1, \ldots, D)$ 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 $\vec{\boldsymbol{\mu}}_\lambda$ and covariance matrices $\boldsymbol{\Sigma}_\lambda$ for each pixel class $\lambda \in \Lambda$.

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

$$\mathcal{L} = \frac{1}{D}\sum_{i=1}^{D} \log\left(\sum_{\lambda\in\Lambda} P(\vec{\boldsymbol{d}}_i \mid \lambda) P(\lambda) \right) \tag{33}$$

The underlying model is that the *complete data* includes not only the observable $\vec{\boldsymbol{d}}_i$ but also the *hidden data* labels $\vec{\boldsymbol{\ell}}_i$ specifying which Gaussian process generated the data $\vec{\boldsymbol{d}}_i$. Actually, $\vec{\boldsymbol{\ell}}_i$ is also a vector of dimension L and $\vec{\boldsymbol{\ell}}_i^{\lambda} = 1$ if $\vec{\boldsymbol{d}}_i$ belongs to class λ 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 2 (EM for Gaussian mixture identification):

① **[Estimation]** Replace $\vec{\boldsymbol{\ell}}_i$ 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:

$$P(\lambda \mid \vec{\boldsymbol{d}}_i) = \frac{P(\vec{\boldsymbol{d}}_i \mid \lambda)P(\lambda)}{\sum_{\lambda \in \Lambda} P(\vec{\boldsymbol{d}}_i \mid \lambda)P(\lambda)}, \tag{34}$$

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

② **[Maximization]** Then, using the current expectation of the labels $\vec{\boldsymbol{\ell}}_i$ as the current labeling of the data, the estimation of the parameters is simple:

$$P(\lambda) = \frac{K_\lambda}{D} \tag{35}$$

$$\vec{\boldsymbol{\mu}}_\lambda = \frac{1}{K_\lambda} \sum_{i=1}^{D} P(\lambda \mid \vec{\boldsymbol{d}}_i)\vec{\boldsymbol{d}}_i \tag{36}$$

$$\boldsymbol{\Sigma}_\lambda = \frac{1}{K_\lambda} \sum_{i=1}^{D} P(\lambda \mid \vec{\boldsymbol{d}}_i)(\vec{\boldsymbol{d}}_i - \vec{\boldsymbol{\mu}}_\lambda)^T(\vec{\boldsymbol{d}}_i - \vec{\boldsymbol{\mu}}_\lambda) \tag{37}$$

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

③ Go to Step ① until convergence. Each iteration is guaranteed to increase the likelihood of the estimates. 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}).

The proposed algorithm has been tested on a variety of color images. We compared segmentation results using color-only, texture-only and combined (color+texture) features [12], [13] and 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. Results has also been compared to those obtained by the JSEG algorithm [65], a recent unsupervised method for color textured image segmentation. Our method clearly outperforms JSEG (see Fig. 8) but JSEG's advantage is that we do not have to specify the image dependent parameter L.

V. Segmentation of Color Images via Reversible Jump MCMC Sampling

Our problem becomes much harder when the number of labels L is also unknown. We have addressed this problem in the context of color-based image segmentation [14], [15]. When this parameter is also being estimated, the unsupervised segmentation problem may be treated as a *model selection* problem over a combined model space. From this point of view, L becomes a *model indicator* and the observation $\mathcal{F}$ is regarded as a three-variate Normal *mixture* with L components corresponding to clusters of pixels which are homogeneous in color.

The goal of our analysis is inference about the number L of Gaussian mixture components (each one corresponds to a label), the component parameters $\Theta = \{\Theta_\lambda = (\vec{\boldsymbol{\mu}}_\lambda, \Sigma_\lambda) \mid \lambda \in \Lambda\}$, the component weights

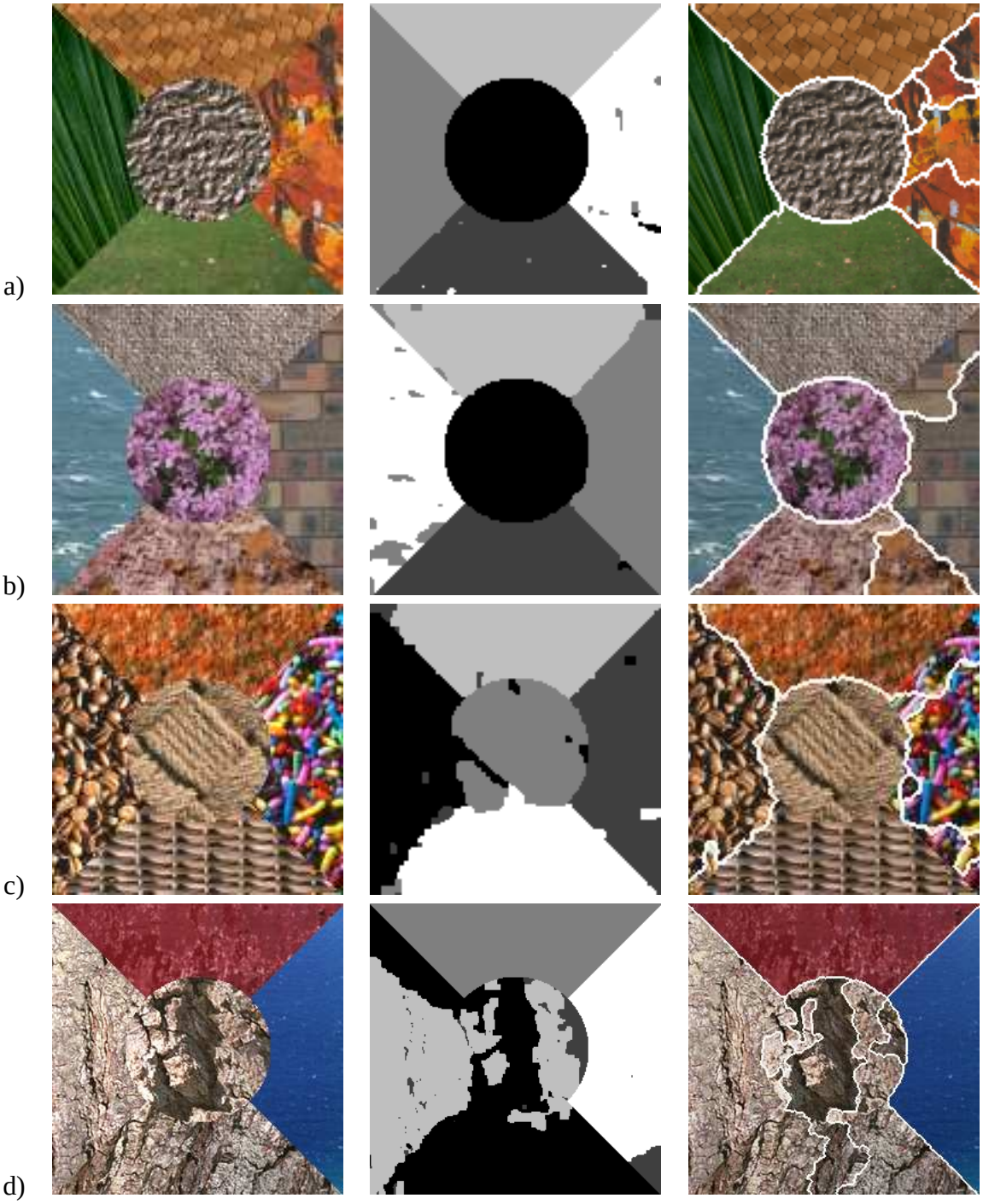


Fig. 8. Unsupervised segmentation results on color textured images, each with 5 classes [13].

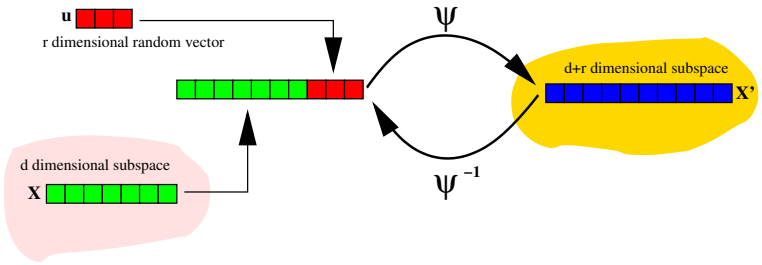


Fig. 9. ψ is a *diffeomorphism* which transforms back and forth between parameter subspaces of different dimensionality [14], [15]. *Dimension matching* can be implemented by generating a random vector u such that the dimensions of (X, u) and X' are equal.

p_λ summing to 1, the inter-pixel interaction strength β, and the segmentation ω. A broadly used tool to sample from the posterior distribution is the Metropolis-Hastings method. Classical methods, however, can not be used due to the changing dimensionality of the parameter space. To overcome this limitation, a promising approach, called Reversible Jump MCMC (RJMCMC), has been adopted [14], [15]. When we have multiple parameter subspaces of different dimensionality, it is necessary to devise different *move types* between the subspaces. These will be combined in a so called *hybrid sampler*. For the color image segmentation model, the following move types are needed [14], [15]:

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

The only randomness in scanning these move types is the random choice between splitting and merging in move (5). One iteration of the hybrid sampler, also called a *sweep*, consists of a complete pass over these moves. The first four move types are conventional in the sense that they do not alter the dimension of the parameter space. Hereafter, we will focus on move (5), which requires the use of the reversible jump mechanism. This move type involves changing L by 1 and making necessary corresponding changes to ω, Θ and p.

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

$$(\Theta^+, p^+) = \psi(\Theta, p, u) \tag{38}$$

where the superscript $+$ denotes parameter vectors after incrementing L. u is a set of random variables having as many elements as the degree of freedom of joint variation of the current parameters (Θ, p) and the proposal (Θ^+, p^+). Note that this definition satisfies the *dimension matching* constraint (see Fig. 9), which guarantees that one can jump back and forth between different parameter sub-spaces [14], [15]. This is needed to ensure the convergence of simulated annealing towards a global optimum. The new parameters of λ_1 and λ_2 are assigned by matching the 0^{th}, 1^{th}, 2^{th} moments of the component being split to those of a combination of the two new components [14], [15]:

$$p_\lambda = p^+_{\lambda_1} + p^+_{\lambda_2} \tag{39}$$

$$p_\lambda \vec{\boldsymbol{\mu}}_\lambda = p^+_{\lambda_1} \vec{\boldsymbol{\mu}}^+_{\lambda_1} + p^+_{\lambda_2} \vec{\boldsymbol{\mu}}^+_{\lambda_2} \tag{40}$$

$$p_\lambda (\vec{\boldsymbol{\mu}}_\lambda \vec{\boldsymbol{\mu}}^T_\lambda + \boldsymbol{\Sigma}_\lambda) = p^+_{\lambda_1} (\vec{\boldsymbol{\mu}}^+_{\lambda_1} \vec{\boldsymbol{\mu}}^{+T}_{\lambda_1} + \boldsymbol{\Sigma}^+_{\lambda_1}) + p^+_{\lambda_2} (\vec{\boldsymbol{\mu}}^+_{\lambda_2} \vec{\boldsymbol{\mu}}^{+T}_{\lambda_2} + \boldsymbol{\Sigma}^+_{\lambda_2}) \tag{41}$$

There are 10 degrees of freedom in splitting λ since covariance matrices are symmetric. Therefore, we need to generate a random variable $u1$, a random vector $\vec{\boldsymbol{u2}}$ and a symmetric random matrix $\boldsymbol{u3}$. We

Original image Segmentation (3 labels) Initial Gaussians Final (3 classes)

Fig. 10. Segmentation of image *rose41* and the estimated Gaussian mixture [14], [15].

can now define the diffeomorphism ψ which transforms the old parameters (Θ, p) into the new (Θ^+, p^+) using the above moment equations and the random numbers $u1$, $\vec{\boldsymbol{u2}}$, and $\boldsymbol{u3}$ [14], [15]:

$$p^+_{\lambda_1} = p_\lambda u1 \tag{42}$$

$$p^+_{\lambda_2} = p_\lambda (1-u1) \tag{43}$$

$$\mu^+_{\lambda_1,i} = \mu_{\lambda,i} + u2_i \sqrt{\Sigma_{\lambda,i,i} \frac{1-u1}{u1}} \tag{44}$$

$$\mu^+_{\lambda_2,i} = \mu_{\lambda,i} - u2_i \sqrt{\Sigma_{\lambda,i,i} \frac{u1}{1-u1}} \tag{45}$$

$$\Sigma^+_{\lambda_1,i,j} = \begin{cases} u3_{i,i} \left(1-{u2_i}^2\right) \Sigma_{\lambda,i,i} \dfrac{1}{u1} & \text{if } i=j \\ u3_{i,j} \Sigma_{\lambda,i,j} \sqrt{\left(1-{u2_i}^2\right)} \sqrt{\left(1-{u2_j}^2\right) u3_{i,i} u3_{j,j}} & \text{if } i \neq j \end{cases} \tag{46}$$

$$\Sigma^+_{\lambda_2,i,j} = \begin{cases} \left(1-u3_{i,i}\right) \left(1-{u2_i}^2\right) \Sigma_{\lambda,i,i} \dfrac{1}{u1} & \text{if } i=j \\ \left(1-u3_{i,j}\right) \Sigma_{\lambda,i,j} \sqrt{\left(1-{u2_i}^2\right)\left(1-{u2_j}^2\right)} \sqrt{\left(1-u3_{i,i}\right)\left(1-u3_{j,j}\right)} & \text{if } i \neq j \end{cases} \tag{47}$$

The random variables u are chosen from the interval $(0, 1]$. In order to favor splitting a class into roughly equal portions, beta$(1.1, 1.1)$ distributions are used. The next step is the reallocation of those sites s where $\omega_s = \lambda$. This reallocation is based on the new parameters and has to be completed in such a way as to ensure the resulting labeling ω^+ is drawn from the posterior distribution with $\Theta = \Theta^+$, $p = p^+$ and $L = L + 1$.

Merging of a pair (λ_1, λ_2) is basically the inverse of the split operation [14], [15].

Finally, the split or merge proposal is accepted with a probability relative to the probability ratio of the current and the proposed states. The segmentation and parameter estimation is then obtained as a MAP estimation implemented via simulated annealing:

Algorithm 3 (RJMCMC Segmentation):

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

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

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

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

The proposed algorithm has been tested [14], [15] on a variety of real color images and results have also been compared to those produced by JSEG [65]. In Fig. 11, we show a couple of results obtained on the Berkeley Segmentation Dataset, and in Fig. 12, we plot the corresponding precision-recall curves. Note that RJMCMC has a slightly higher *F-measure* which ranks it over JSEG. However, it is fair to say that both method perform equally well but behave differently: while JSEG tends to smooth out fine

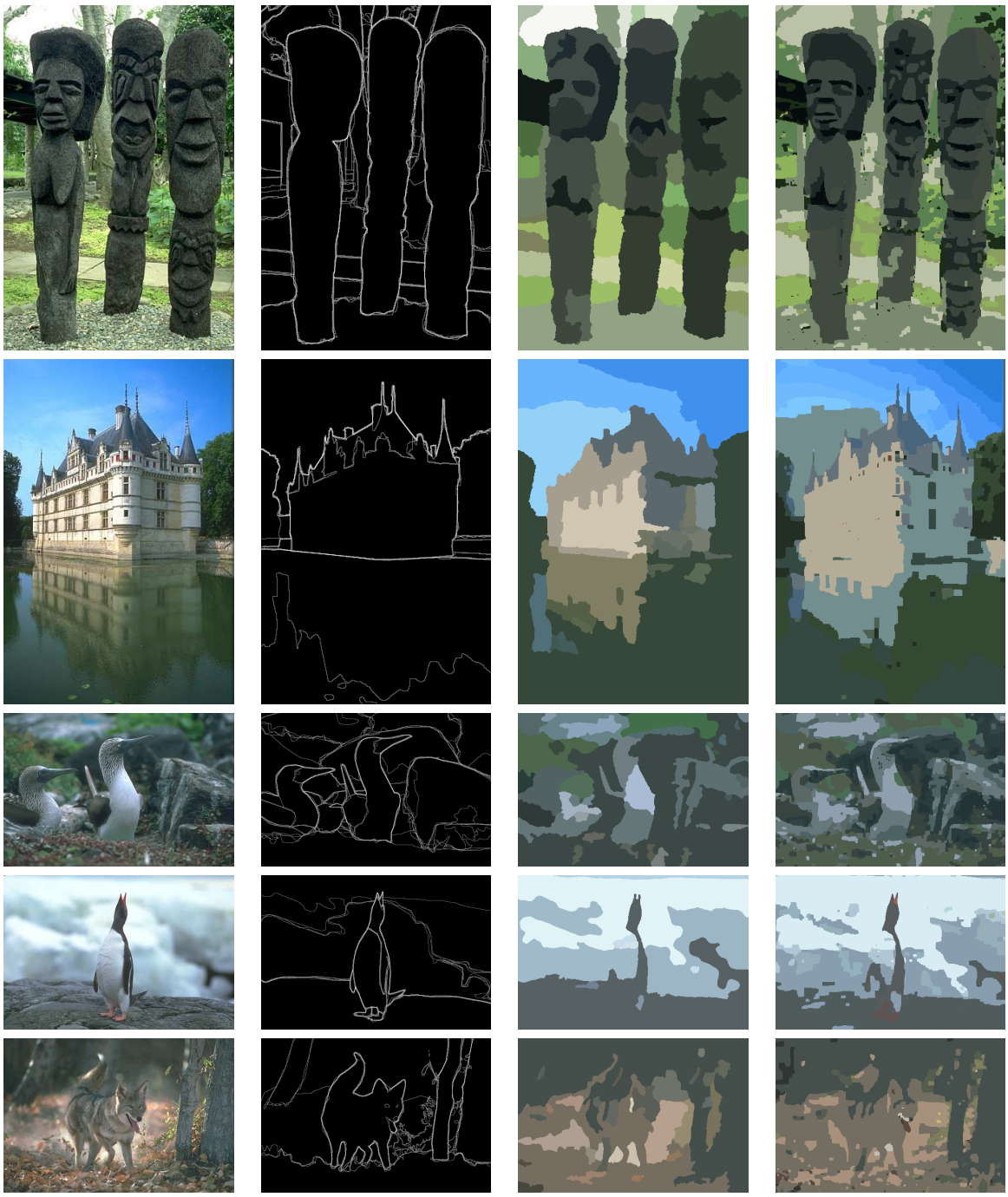
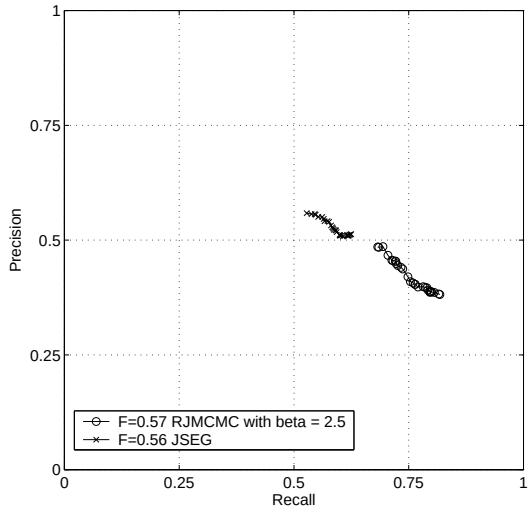


Fig. 11. Benchmark results on images from the Berkeley Segmentation Dataset [15]



Method	F-measure	CPU time
Human segmentation	0.79	—
RJMCMC	0.57	15 min
JSEG	0.56	2 min

Fig. 12. Precision-recall curve, F-measure and CPU time comparison for JSEG and RJMCMC [15].

details (hence it has a higher precision but lower recall value), RJMCMC prefers to keep fine details at the price of producing more edges (*i.e.* its recall values are higher at a lower precision value).

VI. Multilayer MRF Modelization

The human visual system is not treating different features sequentially. Instead, multiple cues are perceived simultaneously and then they are integrated by our visual system in order to explain the observations. Therefore different image features has to be handled in a parallel fashion. We have developed such a model in a Markovian framework and successfully applied it to color-texture [16], [17] and color-motion segmentation [18]–[21]. Herein, we present the MRF image segmentation model which aims at combining color and motion features for video object segmentation [18], [19]. The model has a multi-layer structure (see Fig. 13): 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 each feature layer and provides the segmentation based on the combination of different features. Unlike previous methods, our approach doesn't assume motion boundaries being part of spatial ones. The uniqueness of the proposed method is the ability to detect boundaries that are visible only in the motion feature as well as those visible only in the color one.

Perceptually uniform color values and precomputed optical flow data is used as features for the segmentation. The proposed model consists of 3 layers. At each layer, we use a first order neighborhood system and extra inter-layer cliques (Fig. 13). The image features are represented by multivariate Gaussian distributions. For example, on the color layer, the observed image $\mathcal{F}^c = \{\vec{\boldsymbol{f}}^c_s | s \in \mathcal{S}^c\}$ consists of three spectral component values ($L^*u^*v^*$) at each pixel s denoted by the vector $\vec{\boldsymbol{f}}^c_s$. The class label assigned to a site s on the color layer is denoted by ψ_s. The energy function $U(\psi, \mathcal{F}^c)$ of the so defined MRF layer has the following form:

$$\sum_{s\in\mathcal{S}^c} \mathcal{G}^c(\vec{\boldsymbol{f}}^c_s, \psi_s) + \beta \sum_{\{s,r\}\in\mathcal{C}} \delta(\psi_s, \psi_r) + \sum_{s\in\mathcal{S}^c} V^c(\psi_s, \eta^c_s)$$

where $\mathcal{G}^c(\vec{\boldsymbol{f}}^c_s, \psi_s)$ denotes the Gaussian energy term. The last term ($V^c(\psi_s, \eta^c_s)$) is the inter-layer clique potential. The motion layer adopts a similar energy function with some obvious substitutions (*i.e.* for simplicity, we assume a translational motion model here – for a more elaborate model see [19]).

The combined layer only uses the motion and color features indirectly, through inter-layer cliques. A label consists of a pair of color and motion labels such that $\eta = \langle \eta^c, \eta^m \rangle$, where $\eta^c \in \Lambda^c$ and $\eta^m \in \Lambda^m$. The set of labels is denoted by $\Lambda^x = \Lambda^c \times \Lambda^m$ and the number of classes $L^x = L^c L^m$. Obviously, not

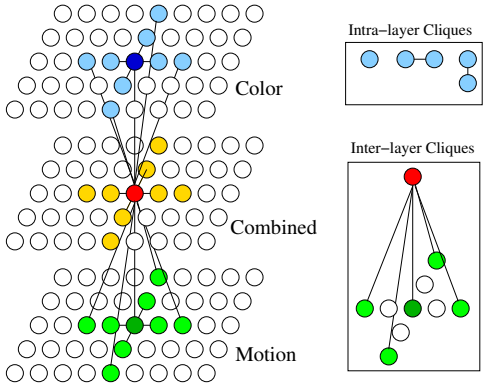


Fig. 13. Multi-layer MRF model [18], [19].

all of these labels are valid for a given image. Therefore the combined layer model also estimates the number of classes and chose those pairs of motion and color labels which are actually present in a given image. The energy function $U(\eta)$ is of the following form:

$$\sum_{s\in\mathcal{S}^x}(V_s(\eta_s)+V^c(\psi_s,\eta_s^c)+V^m(\phi_s,\eta_s^m))+\alpha\sum_{\{s,r\}\in\mathcal{C}}\delta(\eta_s,\eta_r)$$

where $V_s(\eta_s)$ denotes singleton energies, $V^c(\psi_s,\eta_s^c)$ (resp. $V^m(\phi_s,\eta_s^m)$ denotes inter-layer clique potentials. The last term corresponds to second order intra-layer cliques which ensures homogeneity of the combined layer. α has the same role as β in the color layer model and $\delta(\eta_s,\eta_r)=-1$ if $\eta_s=\eta_r$, 0 if $\eta_s\neq\eta_r$ and 1 if $\eta_s^c=\eta_r^c$ *and* $\eta_s^m\neq\eta_r^m$ or $\eta_s^c\neq\eta_r^c$ *and* $\eta_s^m=\eta_r^m$. The idea is that region boundaries present at both color and motion layers are preferred over edges that are found only at one of the feature layers. At any site s, we have 5 inter-layer interactions between two layers: Site s interacts with the corresponding site on the other layer as well as with the 4 neighboring sites two steps away (see Fig. 13). This potential is based on the difference of the first order potentials at the corresponding feature layers. Clearly, the difference is 0 if and only if both the feature layer and the combined layer has the same label. If the labels are different then it is proportional to the energy difference between the two labels. Finally, the singleton energy controls the number of classes at the combined layer by penalizing small classes.

The proposed algorithm has been tested on real video sequences [18], [19]. We also compare the results to motion only and color only segmentation (basically a monogrid model similar to the one defined for the feature layers but without inter-layer cliques). The mean vectors and covariance matrices were computed over representative regions selected by the user. The number of motion and color classes is known a priori but classes on the combined layer are estimated during the segmentation process. Fig. 14 shows some segmentation results. Note that the head of the men on this image can only be separated from the background using motion features. Clearly, the multi-layer model provides significantly better results compared to color only and motion only segmentations. See Fig. 15 to compare the performance of the proposed method with the one from [66] on the *Mother and Daughter* standard sequence. Some of the contours are lost by [66] (the sofa, for example) while our method successfully identifies region boundaries. In particular, our method is able to separate the hand of the mother from the face of the daughter in spite of their similar color. This demonstrates again that the proposed method is quite powerful in combining motion and color features in order to detect boundaries visible only in one of the features. We can also handle occlusion and more complex motions using a modified multilayer model presented in [19]. The model has also been successfully applied to color-textured image segmentation [16], [17] as well as to change detection in aerial images [20], [21].

VII. Conclusion

In this thesis, we have summarized our main contributions to MRF image modeling. We have addressed many aspects of MRF modeling: efficient prior models (multiscale, hierarchical, and multi-layer models);

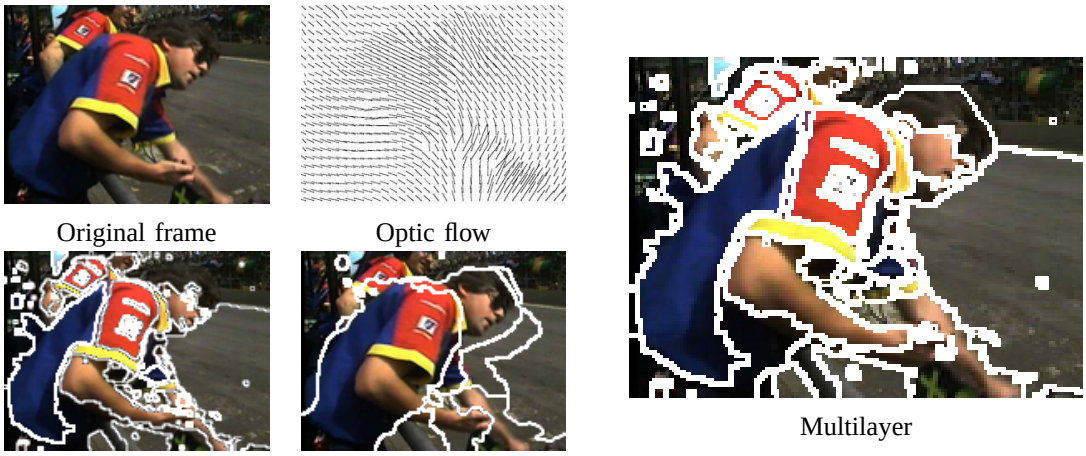


Fig. 14. Segmentation results [18], [19].

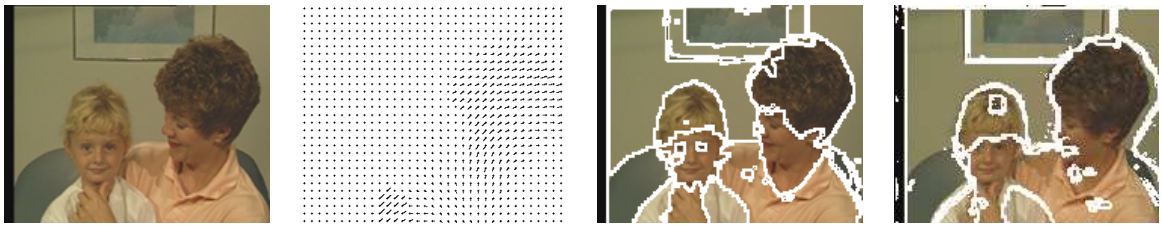


Fig. 15. Comparison of the segmentation results obtained by the proposed method [18], [19] and those produced by the algorithm of Khan & Shah [66].

imaging models for color, texture, motion features, and efficient combination of these segmentation cues; methods to estimate model parameters as well as different optimization techniques. The proposed methods have been applied to a variety of image segmentation problems including remote sensing imagery and change detection. The proposed algorithms proved to be efficient for implementation on special hardware (like the CNN) as well as on parallel architectures.

It is clear, however, that segmentation based solely on low-level image features is a hard problem. Using a classical *smoothness* prior may not be sufficient to achieve good quality results under certain conditions like blur, high noise or cluttered background. In such situations one needs to adopt more elaborate *shape priors*. Indeed, when the form of the regions to be segmented are known a priori (just like for the human visual system) then dealing with blur or noise becomes much easier. This naturally leads to an interest in shape modelling and registration techniques – our current research topics.

REFERENCES

[1] Z. Kato, M. Berthod, and J. Zerubia, "Parallel Image Classification using Multiscale Markov Random Fields." in *Proc. ICASSP*, Minneapolis, Apr. 1993.

[2] ——, "Multiscale Markov random field models for parallel image classification," in *Proceedings of International Conference on Computer Vision*. Berlin, Germany: IEEE, May 1993, pp. 253–257.

[3] ——, "A hierarchical Markov random field model and multi-temperature annealing for parallel image classification," INRIA, Sophia Antipolis, France, Research Report 1938, Aug. 1993. [Online]. Available: http://www.inria.fr/rrrt/rr-1938.html

[4] ——, "Parallel image classification using multiscale Markov random fields," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 5. Minneapolis, USA: IEEE, Apr. 1993, pp. 137–140.

[5] ——, "A hierarchical Markov random field model for image classification," in *Proceedings of International Workshop on Image and Multidimensional Digital Signal Processing*. Cannes, France: IEEE, Sept. 1993.

[6] ——, "A hierarchical Markov random field model and multi-temperature annealing for parallel image classification," *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, vol. 58, no. 1, pp. 18–37, Jan. 1996.

[7] J. Zerubia, Z. Kato, and M. Berthod, "Multi-temperature annealing: A new approach for the energy-minimization of hierarchical Markov random field models," in *Proceedings of International Conference on Pattern Recognition*, vol. 1, IAPR. Jerusalem, Israel: IEEE, Oct. 1994, pp. 520–522.
[8] Z. Kato, J. Zerubia, and M. Berthod, "Unsupervised parallel image classification using a hierarchical Markovian model," INRIA, Sophia Antipolis, France, Research Report 2528, Apr. 1995. [Online]. Available: http://www.inria.fr/rrrt/rr-2528.html
[9] Z. Kato, M. Berthod, J. Zerubia, and W. Pieczynski, "Unsupervised adaptive image segmentation," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 4. Detroit, Michigan, USA: IEEE, May 1995, pp. 2399–2402.
[10] Z. Kato, J. Zerubia, and M. Berthod, "Unsupervised parallel image classification using a hierarchical Markovian model," in *Proceedings of International Conference on Computer Vision*. Cambridge, MA, USA: IEEE, June 1995, pp. 169–174.
[11] ——, "Unsupervised parallel image classification using Markovian models," *Pattern Recognition*, vol. 32, no. 4, pp. 591–604, Apr. 1999.
[12] Z. Kato and T. C. Pong, "A Markov random field image segmentation model using combined color and texture features," in *Proceedings of International Conference on Computer Analysis of Images and Patterns*, ser. Lecture Notes in Computer Science, W. Skarbek, Ed., vol. 2124. Warsaw, Poland: Springer, Sept. 2001, pp. 547–554.
[13] ——, "A Markov random field image segmentation model for color textured images," *Image and Vision Computing*, vol. 24, no. 10, pp. 1103–1114, Oct. 2006.
[14] Z. Kato, "Reversible jump Markov chain Monte Carlo for unsupervised MRF color image segmentation," in *Proceedings of British Machine Vision Conference*, A. Hoppe, S. Barman, and T. Ellis, Eds., vol. 1. Kingston, UK: BMVA, Sept. 2004, pp. 37–46.
[15] ——, "Segmentation of color images via reversible jump MCMC sampling," *Image and Vision Computing*, 2007, in press.
[16] Z. Kato, T. C. Pong, and G. Q. Song, "Multicue MRF image segmentation: Combining texture and color," in *Proceedings of International Conference on Pattern Recognition*, vol. 1, IAPR. Quebec, Canada: IEEE, Aug. 2002, pp. 660–663.
[17] ——, "Unsupervised segmentation of color textured images using a multi-layer MRF model," in *Proceedings of International Conference on Image Processing*, vol. I. Barcelona, Spain: IEEE, Sept. 2003, pp. 961–964.
[18] Z. Kato and T. C. Pong, "Video object segmentation using a multicue Markovian model," in *Joint Hungarian-Austrian Conference on Image Processing and Pattern Recognition*, D. Chetverikov, L. Czuni, and M. Vincze, Eds., KEPAF, OAGM/AAPR. Veszprem, Hungary: Austrian Computer Society, May 2005, pp. 111–118.
[19] ——, "A multi-layer MRF model for video object segmentation," in *Proceedings of Asian Conference on Computer Vision*, ser. Lecture Notes in Computer Science, P. J. Narayanan, S. K. Nayar, and H.-Y. Shum, Eds., vol. 3852. Hyderabad, India: Springer, Jan. 2006, pp. 953–962.
[20] C. Benedek, T. Sziranyi, Z. Kato, and J. Zerubia, "A multi-layer MRF model for object-motion detection in uregistered airborne image-pairs," in *Proceedings of International Conference on Image Processing*, IEEE. San Antonio, Texas, USA: IEEE, Sept. 2007.
[21] C. Benedek, T. Szirnyi, Z. Kato, and J. Zerubia, "A three-layer MRF model for object motion detection in airborne images," INRIA, Sophia Antipolis, France, Research Report 6208, June 2007. [Online]. Available: https://hal.inria.fr/inria-00150805
[22] Z. Kato, J. Zerubia, and M. Berthod, "Image classification using Markov random fields with two new relaxation methods: Deterministic pseudo annealing and modified Metropolis dynamics," INRIA, Sophia Antipolis, France, Research Report 1606, Feb. 1992. [Online]. Available: http://www.inria.fr/rrrt/rr-1606.html
[23] ——, "Satellite image classification using a modified Metropolis dynamics," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 3. San-Francisco, California, USA: IEEE, Mar. 1992, pp. 573–576.
[24] ——, "Bayesian image classification using Markov random fields," in *Maximum Entropy and Bayesian Methods*, A. Mohammad-Djafari and G. Demoment, Eds. Kluwer Academic Publisher, 1993, pp. 375–382.
[25] M. Berthod, Z. Kato, and J. Zerubia, "DPA: A deterministic approach to the MAP," *IEEE Transactions on Image Processing*, vol. 4, no. 9, pp. 1312–1314, Sept. 1995.
[26] M. Berthod, Z. Kato, S. Yu, and J. Zerubia, "Bayesian image classification using Markov random fields," *Image and Vision Computing*, vol. 14, pp. 285–295, 1996.
[27] C. Gradwohl and Z. Kato, "Supervised image segmentation using Markov random fields," Demo program, Feb 2005. [Online]. Available: http://www.inf.u-szeged.hu/~kato/software/mrfdemo.html
[28] T. Sziranyi, J. Zerubia, D. Geldreich, and Z. Kato, "Cellular neural network in Markov random field image segmentation," in *Proceedings of International Workshop on Cellular Neural Networks and their Applications*. Seville, Spain: IEEE, June 1996, pp. 139–144.
[29] T. Sziranyi, J. Zerubia, L. Czuni, D. Geldreich, and Z. Kato, "Image segmentation using Markov random field model in fully parallel cellular network architectures," *Real Time Imaging*, vol. 6, no. 3, pp. 195–211, June 2000.
[30] Z. Kato, T. C. Pong, and J. C. M. Lee, "Color image segmentation and parameter estimation in a Markovian framework," *Pattern Recognition Letters*, vol. 22, no. 3-4, pp. 309–321, Mar. 2001.
[31] ——, "Motion compensated color video classification using Markov random fields," in *Proceedings of Asian Conference on Computer Vision*, ser. Lecture Notes in Computer Science, R. Chin and T. C. Pong, Eds., vol. 1351. Hong Kong, China: Springer, Jan. 1998, pp. 738–745.
[32] Z. Kato, X. Ji, T. Sziranyi, Z. Toth, and L. Czuni, "Content-based image retrieval using stochastic paintbrush transformation," in *Proceedings of International Conference on Image Processing*, vol. 1. New York, USA: IEEE, Sept. 2002, pp. 944–947.
[33] X. Ji, Z. Kato, and Z. Huang, "Non-photorealistic rendering and content-based image retrieval," in *Proceedings of Pacific Conference on Computer Graphics and Applications*. Canmore, Canada: IEEE, Oct. 2003, pp. 153–162.
[34] D. Cremers, F. Tischhauser, J. Weickert, and C. Schnorr, "Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 295–313, 2002.
[35] C. Kervrann and F. Heitz, "Statistical deformable model-based segmentation of image motion," *IEEE Transactions on Image Processing*, vol. 8, pp. 583–588, 1999.
[36] M. Rochery, I. H. Jermyn, and J. Zerubia, "Higher order active contours and their application to the detection of line networks in

satellite imagery," in *Proceedings of* 2^{nd} *IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, at ICCV, Nice, France, Oct. 2003.
[37] Z. Tu, X. Chen, A. Yuille, and S.-C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *International Journal of Computer Vision*, no. Marr Prize Issue, 2005.
[38] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.
[39] S. A. Barker and P. J. W. Rayner, "Unsupervised image segmentation using Markov random field models," *Pattern Recognition*, vol. 33, no. 4, pp. 587–602, Apr 2000.
[40] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of gibbs random fields using simulated annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 799–813, Aug. 1989.
[41] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.
[42] A. Blake and A. Zisserman, *Visual reconstruction*. MIT Press, 1987.
[43] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577–685, 1989.
[44] D. Kersten, P. Mamassian, and A. Yuille, "Object perception as Bayesian inference," *Annual Review of Psychology*, vol. 55, pp. 271–304, 2004.
[45] D. Mumford, "Pattern theory: a unifying perspective," in *Perception as Bayesian Inference*, D. Knill and W. Richards, Eds. Cambridge University Press, 1996, pp. 25–62.
[46] B. Chalmond, *Modeling and Inverse Problems in Image Analysis*. Springer, 2003.
[47] G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*, 2nd ed. Springer, 2003.
[48] D. Mumford, "The Bayesian rationale for energy functionals," in *Geometry-Driven Diffusion in Computer Vision*, B. Romeny, Ed. Kluwer Academic, 1994, pp. 141–153.
[49] N. Vandenbroucke, L. Macaire, and 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.
[50] J. C. Noordam, G. W. Otten, A. J. M. Timmermans, and B. v. Zwol, "High-speed potato grading and quality inspection based on a color vision system," in *Proc. of SPIE Machine Vision Applications in Industrial Inspection*, K. W. T. Jr., Ed., vol. 3966, 2000, pp. 206–220.
[51] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Statist. Soc., ser. B*, vol. vol. 39, no. 1, pp. 1–38, 1977.
[52] G. Celeux and J. Diebolt, "The SEM algorithm: a probabilisitic teacher algorithm derived from the EM algorithm for the mixture problem," *Computational Statistics Quaterly*, vol. 2, pp. 73–82, 1985.
[53] 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.
[54] B. Chalmond, "An Iterative Gibbsian Technique for Reconstruction of M-ary Images," *Pattern Recognition*, vol. 22, no. 6, 1989.
[55] B. Braathen, W. Pieczynski, and P. Masson, "Global and Local Methods of Unsupervised Bayesian Segmentation of Images," *Machine Graphics and Vision*, vol. 2, no. 1, pp. 39–52, 1993.
[56] N. Giordana and W. Pieczynski, "Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 465–475, May 1997.
[57] 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.
[58] L. Gupta and T. Sortrakul, "A Gaussian-mixture-based image segmentation algorithm," *Pattern Recognition*, vol. 31, no. 3, pp. 315–325, 1998.
[59] R. Azencott, "Markov fields and image analysis," *Proc. AFCET, Antibes*, 1987.
[60] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society, series B*, vol. 48, no. 3, pp. 259–302, 1986.
[61] P. V. Laarhoven and E. Aarts, "Simulated annealing : Theory and applications," *Reidel Pub., Dordrecht, Holland*, 1987.
[62] Y. Bard, *Nonlinear Parameter Estimation*. Academic Press, Inc., 1974.
[63] W. Pieczynski, "Statistical image segmentation," in *Machine Graphics and Vision, GKPO'92*, Naleczow, Poland, May 1992, pp. 261–268.
[64] D. Geman, "Bayesian image analysis by adaptive annealing," in *Proc. IGARSS'85*, Amherst, USA, Oct. 1985, pp. 269–277.
[65] Y. Deng, , and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, Aug. 2001. [Online]. Available: http://vision.ece.ucsb.edu/segmentation/jseg/
[66] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, vol. II. Kauai, Hawaii: IEEE, Dec. 2001, pp. 746–751.