PERGAMON

Pattern Recognition 32 (1999) 591–604

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

Unsupervised parallel image classification using Markovian models[1]

Zoltan Kato[2], Josiane Zerubia*, Marc Berthod

INRIA - 2004 Route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex, France

Received 27 October 1997; received in revised form 12 June 1998

Abstract

This paper deals with the problem of unsupervised classification of images modeled by Markov random fields (MRF). If the model parameters are known then we have various methods to solve the segmentation problem (simulated annealing (SA), iterated conditional modes (ICM), etc). However, when the parameters are unknown, the problem becomes more difficult. One has to estimate the hidden label field parameters only from the observed image. Herein, we are interested in parameter estimation methods related to monogrid and hierarchical MRF models. The basic idea is similar to the expectation–maximization (EM) 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 at the previous step. The only parameter supposed to be known is the number of classes, all the other parameters are estimated. The proposed algorithms have been implemented on a Connection Machine CM200. Comparative experiments have been performed on both noisy synthetic data and real images.

Keywords: Markov random field model; Hierarchical model; Parameter estimation; Parallel unsupervised image classification

1. Introduction

Image classification 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, indexing, for example) are based on such a preprocessing. Our approach consists in building a probabilistic model and finding the most likely labeling (or classification). To do so, we need to define some probability measure on the set of all possible labelings. In real images, neighboring pixels usually have similar properties. Within a probabilistic framework, such regularities are well expressed by means of MRF. Another reason for dealing with MRF models is the *Hammersley–Clifford theorem*, which allows to define MRFs through clique-potentials.

In real-life applications, clique-potentials are usually unknown, one has to estimate [1, 2] them only from the observed image. From a statistical viewpoint, this means that we want to estimate parameters from random variables whose joint distribution is a mixture of distributions. If we have a realization of the label field then the

*Corresponding author. Tel.: + 33 4 92.38.78.57; fax: + 33 4 92.38.76.43; e-mail: zerubia@sophia.inria.fr

[1]This work has been partially funded by CNES (French Space Agency), AFIRST and DRED/GdRISIS.

[2]Now at CWI-P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands

0031-3203/99/$ – See front matter
PII: S0031-3203(98)00104-6

problem is relatively easy, many classical methods are available to do such a parameter estimation (maximum likelihood, coding method [3] etc). Unfortunately, such a realization is usually unknown, so the direct use of the above mentioned algorithms is impossible. Therefore, we have to work with incomplete data methods.

Some algorithms are iterative, [4–6] generating a labeling, estimating parameters from it, then generating a new labeling using these parameters, etc. For such methods, we need a reasonably good initial value for each parameter. Since the classes are represented by Gaussian distributions in the models considered herein, the initialization of the mean and the variance of each class is very important because of the influence of such initial conditions on subsequent labelings and hence on the quality of the final estimates. This problem is related to the determination of the modes of a Gaussian mixture without any a priori information. Many techniques are available: Method of moments [7], Prony's method [8], or geometrical analysis of the histogram [9], for instance.

In this paper, we propose parameter estimation methods for both monogrid [10] and hierarchical [11, 12] MRF models. The algorithms described herein have been tested on image segmentation problems. Comparative tests have been conducted on noisy synthetic data and on real satellite images.

In Section 2, we give a general overview of the parameter estimation problem. In Section 3, we present a monogrid and a hierarchical MRF segmentation model and show how to estimate the related parameters. In Section 4, we give some details concerning the parallel implementation of the proposed methods. Finally, in Section 5, we present experimental results obtained on a Connection Machine CM200.

2. The parameter estimation problem

Image labeling is a general framework to solve low-level vision tasks, such as image classification, edge detection, etc. To each pixel of the image, we assign a label. The meaning of the labels depends on the problem that we want to tackle. For image classification, for example, a label means a class; for edge detection, it means the presence or the direction of an edge, etc. Thus, we have to deal with the following general problem:

We are given a set of pixels (an image) $\mathscr{S} = \{s_1, s_2, \ldots, s_N\}$ and $\mathscr{F} = \{f_s : s \in \mathscr{S}\}$ a set of image data (grey levels, for instance). Each of these pixels may take a label from $\Lambda = \{0, 1, \ldots, L-1\}$. The configuration space Ω is the set of all global discrete labelings $\omega = (\omega_{s_1}, \ldots, \omega_{s_N})$, $\omega_s \in \Lambda$. The label process is denoted by $\mathscr{X}$ and it is modeled by a MRF. Furthermore, we are given n parameters forming a vector Θ which appears in the MRF model.

Now, we construct a Bayesian estimator to find the optimal labeling, that is the labeling which maximizes the posterior distribution of the label field:

$$\hat{\omega} = \arg\max_{\omega \in \Omega} P_{\Theta}(\omega | \mathscr{F}, \Theta), \tag{1}$$

where $\hat{\omega}$ is the MAP estimate of the label field, given $\mathscr{F}$, under the model P_{Θ} (hereafter, the index Θ will be omitted). If both Θ and ω are unknown, the maximization problem in Eq. (1) becomes: [13,14]:

$$(\hat{\omega}, \hat{\Theta}) = \arg\max_{\omega, \Theta} P(\omega, \mathscr{F} | \Theta). \tag{2}$$

The pair $(\hat{\omega}, \hat{\Theta})$ is the global maximum of the joint probability $P(\omega, \mathscr{F} | \Theta)$. If we regard Θ as a random variable, the above maximization is an ordinary MAP estimation in the following way [13]. Let us suppose that Θ is restricted to a finite volume domain $\mathscr{D}_{\Theta}$ and Θ is uniform on $\mathscr{D}_{\Theta}$ (that is $P(\Theta)$ is constant). Then, we get [13]:

$$\arg\max_{\omega, \Theta} P(\omega, \Theta | \mathscr{F}) = \arg\max_{\omega, \Theta} \frac{P(\omega, \mathscr{F} | \Theta) P(\Theta)}{P(\mathscr{F})}$$

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

However, this maximization is very difficult, having no direct solution. Even SA is not implementable because the local characteristics with respect to the parameters Θ cannot be computed from $P(\omega, \mathscr{F} | \Theta)$. A possible solution is to adopt the following criterion instead [13,14]:

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

$$\hat{\Theta} = \arg\max_{\Theta} P(\hat{\omega}, \mathscr{F} | \Theta) \tag{5}$$

Of course, the solution of the above equations is not necessarily the joint maximum corresponding to Eq. (3), but in practice it is a good approximation. Clearly, Eq. (4) is equivalent to Eq. (2) for $\Theta = \hat{\Theta}$ and Eq. (5) is equivalent to Eq. (2) with $\omega = \hat{\omega}$. Furthermore, Eq. (4) is equivalent to the MAP estimate of ω in the case of known parameters:

$$\arg\max_{\omega} P(\omega, \mathscr{F} | \hat{\Theta}) = \arg\max_{\omega} P(\omega | \mathscr{F}, \hat{\Theta})$$

$$P(\mathscr{F} | \hat{\Theta}) = \arg\max_{\omega} P(\omega | \mathscr{F}, \hat{\Theta}). \tag{6}$$

Hereafter, we briefly overview two estimation methods that can be used to solve the system Eqs. (4) and (5). In Section 3, we propose two unsupervised image segmentation methods based on these algorithms.

2.1. Adaptive simulated annealing (ASA)

Adaptive Simulated Annealing (ASA) has been proposed by Geman in Ref. [13]. The algorithm was adapted to image segmentation problems in Ref. [14], where the convergence of ASA has also been proved.

Algorithm 2.1. (ASA)

① *Set* $k = 0$ *and initialize* $\hat{\Theta}^0$.
② *Do* n *iterations* ($n \geqslant 1$) *of Gibbs sampling from* $P(\omega|\mathscr{F}, \hat{\Theta}^k)$. *The resulting labeling is denoted by* $\hat{\omega}^{k+1}$.
③ *Update the current estimate of the parameters,* $\hat{\Theta}^{k+1}$ *to the ML estimate based on the current labeling* $\hat{\omega}^{k+1}$.
④ *Goto Step* ② *with* $k = k + 1$ *until* $\hat{\Theta}$ *stabilizes.*

If the ML estimate is not tractable, which is often the case when dealing with MRF models, one can use an approximation (Maximum Pseudo Likelihood (MPL), for instance). We remark that a similar algorithm has been reported in Ref. [3]. It uses ICM instead of the Gibbs Sampler in Step ②. We also use this latter version in Section 3.

2.2. Iterative conditional estimation (ICE)

Another solution to the incomplete data problem has been proposed by Pieczynski et al. [15, 16, 6] Let us consider an estimator $\mathscr{E}_\Theta(\mathscr{F}, \omega)$ of Θ (ML, for instance). Since realizations of the label field are unknown, the direct use of $\mathscr{E}_\Theta(\mathscr{F}, \omega)$ is impossible, we have to approximate it. The best approximation, in the mean-square sense, is the *conditional expectation.* Since $E\{\mathscr{E}_\Theta|\mathscr{F}, \omega\}$ depends on the parameters Θ, we need a parameter $\hat{\Theta}^k$ previously defined in some way. This yields an iterative procedure, called ICE [16,6].

Algorithm 2.2. (ICE)

① *Set* $k = 0$ *and initialize* $\hat{\Theta}^0$.
② *Generate* n *realizations* (n *is a priori chosen*) $\hat{\omega}^i (1 \leqslant i \leqslant n)$ *of the label field based on* $\hat{\Theta}^k$.
③ *Based on the sample* $\hat{\omega}^i (1 \leqslant i \leqslant n)$, $\hat{\Theta}^{k+1}$ *is obtained as the conditional expectation*

$$\hat{\Theta}^{k+1} = E\{\mathscr{E}_\Theta|\mathscr{X} = \omega\} \approx \frac{1}{n}\sum_{i=1}^{n} \mathscr{E}_\Theta(\mathscr{F}, \hat{\omega}^i). \tag{7}$$

④ *Goto Step* ② *until* $\hat{\Theta}$ *stabilizes.*

3. Unsupervised image segmentation

3.1. Monogrid model

Herein, we consider a monogrid MRF segmentation model originally presented in Ref. [10] but with unknown parameters [17]. Let us first review the model. We are given the gray-levels $\mathscr{F}$ of an image $\mathscr{S} = \{s_1, s_2, \ldots, s_N\}$, which is the only observed attribute. Moreover, we are given a set of labels denoted by $\Lambda = \{0, 1, \ldots, L-1\}$. The problem is to estimate the model parameters Θ and find the MAP estimate of the label field $\mathscr{X}$ among all the possible discrete labelings $\Omega = \Lambda^N = \{\omega = (\omega_{s_1}, \ldots, \omega_{s_N}), \omega_s \in \Lambda\}$. As explained before, in the case of unknown parameters, the maximization problem becomes (cf. Eq. (2)):

$$(\hat{\omega}, \hat{\Theta}) = \arg\max_{\omega, \Theta} P(\omega, \mathscr{F}|\Theta). \tag{8}$$

Since this maximization is not tractable, we use Eqs. (4) and (5) instead. The maximization problem in Eq. (4) corresponds to the ordinary MAP estimate with known parameters. Herein, we are interested in the solution of the ML estimation using Eq. (5):

$$\hat{\Theta} = \arg\max_{\Theta} P(\hat{\omega}, \mathscr{F}|\Theta) \tag{9}$$

The probability on the right-hand side can be written as

$$P(\hat{\omega}, \mathscr{F}|\Theta) = P(\mathscr{F}|\hat{\omega}, \Theta)\, P(\hat{\omega}|\Theta) \tag{10}$$

Using the model defined in Ref. [10], the first term is a product of independent Gaussian densities and the second term is a first-order MRF, also known as the Potts model in statistical mechanics [18]:

$$P(\hat{\omega}, \mathscr{F}|\Theta) = \prod_{s\in\mathscr{S}} \frac{1}{\sqrt{2\pi}\sigma_{\hat{\omega}_s}} \exp\left(-\frac{(f_s - \mu_{\hat{\omega}_s})^2}{2\sigma_{\hat{\omega}_s}^2}\right) \times \frac{\exp(-\beta\sum_{\{s,r\}\in C}\delta(\hat{\omega}_s, \hat{\omega}_r))}{Z(\beta)} \tag{11}$$

with

$$Z(\beta) = \sum_{\omega\in\Omega} \exp\left(-\beta \sum_{\{s,r\}\in C} \delta(\omega_s, \omega_r)\right) \tag{12}$$

and

$$\delta(\hat{\omega}_s, \hat{\omega}_r) = \begin{cases} 0 & \text{if } \hat{\omega}_s = \hat{\omega}_r, \\ 1 & \text{otherwise,} \end{cases} \quad \beta > 0. \tag{13}$$

We have $2L + 1$ parameters (two for each class and one hyperparameter β). The first $2L$ parameters are estimated from the Gaussian term and the last one is computed from the Markovian term. Instead of the likelihood function defined in Eq. (11), we consider the simpler logarithmic likelihood:

$$\ln(L(\Theta)) = \sum_{s\in\mathscr{S}} \left(-\ln(\sqrt{2\pi}\sigma_{\hat{\omega}_s}) - \frac{(f_s - \mu_{\hat{\omega}_s})^2}{2\sigma_{\hat{\omega}_s}^2}\right) - \beta \sum_{\{s,r\}\in C} \delta(\hat{\omega}_s, \hat{\omega}_r) - \ln(Z(\beta)) \tag{14}$$

$$= \sum_{\lambda\in\Lambda} \sum_{s\in\mathscr{S}_\lambda} \underbrace{\left(-\ln(\sqrt{2\pi}\sigma_\lambda) - \frac{(f_s - \mu_\lambda)^2}{2\sigma_\lambda^2}\right)}_{\mathscr{G}(\mu_\lambda, \sigma_\lambda)} - \underbrace{\beta \sum_{\{s,r\}\in C} \delta(\hat{\omega}_s, \hat{\omega}_r) - \ln(Z(\beta))}_{\mathscr{M}(\beta)}, \tag{15}$$

where $\mathscr{S}_\lambda$ is the set of pixels where $\hat{\omega} = \lambda$. To get the maximum of the likelihood function, $\hat{\Theta}$ must satisfy the

following equations:

$$\forall\lambda\in\Lambda:\frac{\partial\mathscr{G}(\mu_\lambda,\sigma_\lambda)}{\partial\mu_\lambda}=0, \tag{16}$$

$$\frac{\partial\mathscr{G}(\mu_\lambda,\sigma_\lambda)}{\partial\sigma_\lambda}=0, \tag{17}$$

$$\text{and}\quad\frac{\partial\mathscr{M}(\beta)}{\partial\beta}=0. \tag{18}$$

The solution of the above system for μ_λ and σ_λ is simply the empirical mean and variance:

$$\forall\lambda\in\Lambda:\quad\mu_\lambda=\frac{1}{|\mathscr{S}_\lambda|}\sum_{s\in\mathscr{S}_\lambda}f_s,$$

$$\sigma_\lambda^2=\frac{1}{|\mathscr{S}_\lambda|}\sum_{s\in\mathscr{S}_\lambda}(f_s-\mu_\lambda)^2. \tag{19}$$

The solution for β, however, is not as easy. Let us consider the derivative of $\mathscr{M}(\beta)$:

$$\frac{\partial}{\partial\beta}\left(-\beta N^{ih}(\hat{\omega})-\ln\left(\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega))\right)\right)$$

$$=-N^{ih}(\hat{\omega})+\frac{\sum_{\omega\in\Omega}N^{ih}(\omega)\exp(-\beta N^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega))}=0 \tag{20}$$

with $N^{ih}(\hat{\omega})=\sum_{\{s,r\}\in C}\delta(\hat{\omega}_s,\hat{\omega}_r)$ is the number of inhomogeneous cliques in $\hat{\omega}$. From Eq. [20], we get

$$N^{ih}(\hat{\omega})=\frac{\sum_{\omega\in\Omega}N^{ih}(\omega)\exp(-\beta N^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega))}. \tag{21}$$

The right-hand side of the above equation is also called the *energy mean*. Since $\ln(Z(\beta))$ is *convex* in Θ, [18,13], the gradient can be approximated by stochastic relaxation [13].

Herein, we use a simpler heuristic, which is computationally less expensive and gives reasonably good results, in practice: Suppose that we have an estimate of the label field $\hat{\omega}$. The algorithm aims at finding a $\hat{\beta}$ which does not change the labeling $\hat{\omega}$ during a few iterations of a fixed temperature Metropolis algorithm [17]. The temperature T is chosen empirically on a trial and error basis. In our tests, we have set $T=2.5$. The idea behind $T=2.5$ is that a too high value ($T\geqslant 4$) would result in a completely random labeling independent of $\hat{\omega}$ and the algorithm will not converge. On the other hand, a too small value ($T\leqslant 1$) turns the Metropolis algorithm into a deterministic one, which permits a large variation in $\hat{\beta}$ without really disturbing $\hat{\omega}$. The formulation of the proposed algorithm is the following:

Algorithm 3.1. (Hyperparameter estimation)

① *Set $k=0$, initialize $\hat{\beta}^0$ and let $N^{ih}(\hat{\omega})$ denote the number of inhomogeneous cliques in the labeling estimate.*

② *Using Metropolis algorithm at a fixed temperature T, generate a new labeling η, sampling from*

$$P(\mathscr{X}=\omega)=\frac{\exp(-\frac{\hat{\beta}^k}{T}\sum_{\{s,r\}\in\mathscr{S}}\delta(\omega_s,\omega_r))}{Z(\hat{\beta}^k)}. \tag{22}$$

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

③ *If $N^{ih}(\eta)\approx N^{ih}(\hat{\omega})$ then stop, else $k=k+1$. If $N^{ih}(\eta)<N^{ih}(\hat{\omega})$ then decrease $\hat{\beta}^k$, if $N^{ih}(\eta)>N^{ih}(\hat{\omega})$ then increase $\hat{\beta}^k$, and goto Step ②.*

The complete parameter estimation process is the following:

Algorithm 3.2. (Unsupervised segmentation)

① *Given an image $\mathscr{F}$, initialize β, μ_λ and σ_λ for each $\lambda\in\Lambda$.*

② (Estimation) *Using Algorithm* 2.2 (*ICE*), *get an estimate $\hat{\Theta}$ of the parameters.*

③ (Segmentation) *Given the parameters $\hat{\Theta}$, do an ordinary segmentation with known parameters to get the MAP estimate of the label field given $\mathscr{F}$ and $\hat{\Theta}$.*

3.2. Hierarchical model

First, let us briefly review the hierarchical model proposed in Ref. [12]. In the followings, we suppose that $\mathscr{S}=\{s_1,s_2,\ldots,s_N\}$ is a $W\times H$ lattice, so that:

$$\mathscr{S}\equiv\mathscr{L}=\{(i,j):1\leqslant i\leqslant W\text{ and }1\leqslant j\leqslant H\}, \tag{23}$$

and $W=w^n$, $H=h^m$. This assumption introduces some restrictions on $\mathscr{L}$ but this is not crucial, in practice, since we work mostly on images where both W and H are a power of 2. First, we generate a label pyramid but we keep the whole observation field: For all $1\leqslant i\leqslant M$ ($M=\inf(n,m)$), $\mathscr{S}$ is divided into blocks of size $w^i\times h^i$, denoted by b^i. These blocks will form a coarser scale $\mathscr{P}^i$. The labels assigned to the sites of a block are supposed to be the same over the whole block. Then, a block b^i is "transformed" into a unique site s^i at the corresponding level in the pyramid. We have the same neighborhood structure at coarser grids in the pyramid as on the finest (initial) grid (we note the cliques at level i by C^i). Let $\bar{\mathscr{S}}=\{\bar{s}_1,\ldots,\bar{s}_N\}$ denote the sites of this pyramid. We introduce new interactions between two neighbor grids in the pyramid (see Fig. 1). $\bar{\mathscr{G}}$ denotes this new neighborhood system defined on the whole pyramid. Furthermore, let $\bar{\mathscr{X}}$ be a MRF over $\bar{\mathscr{G}}$ with energy function $\bar{U}$ and potentials $\{\bar{V}_{\bar{C}}\}_{\bar{C}\in\bar{\mathscr{C}}}$. The energy function is of the following form:

$$\bar{U}(\bar{\omega},\mathscr{F})=\bar{U}_1(\bar{\omega},\mathscr{F})+\bar{U}_2(\bar{\omega}), \tag{24}$$

$$\bar{U}_1(\bar{\omega},\mathscr{F})=\sum_{\bar{s}\in\bar{\mathscr{S}}}\bar{V}_1(\bar{\omega}_{\bar{s}},\mathscr{F})=\sum_{i=0}^{M}\sum_{s^i\in S_i}V_1^i(\omega_{s^i}^i,\mathscr{F})$$

$$=\sum_{i=0}^{M}U_1^i(\omega^i,\mathscr{F}), \tag{25}$$

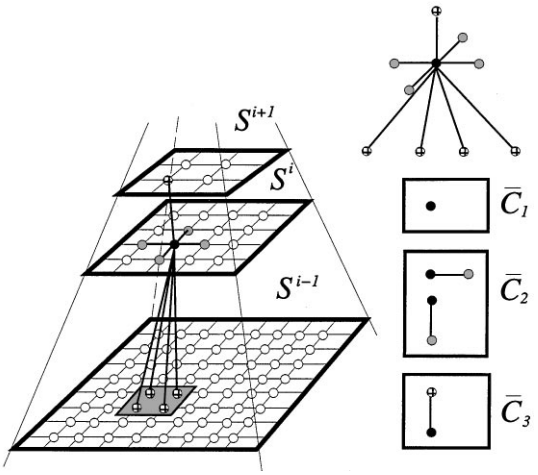

Fig. 1. Hierarchical MRF model.

$$\bar{U}_2(\bar{\omega}) = \sum_{C\in\bar{\mathscr{C}}_2} \bar{V}_2(\bar{\omega}_C) + \sum_{C\in\bar{\mathscr{C}}_3} \bar{V}_2(\bar{\omega}_C)$$
$$= \sum_{i=0}^{M} U_2^i(\omega^i) + \sum_{C\in\bar{\mathscr{C}}_3} \bar{V}_2(\bar{\omega}_C)$$
$$= \sum_{i=0}^{M} \sum_{C\in\mathscr{C}^i} V_2^i(\omega_C^i) + \sum_{C\in\bar{\mathscr{C}}_3} \bar{V}_2(\bar{\omega}_C), \tag{26}$$

where $\bar{\mathscr{C}}_3$ denotes the new cliques siting astride two neighbor grids. $\bar{V}_2(\bar{\omega}_C)$ is the potential function over these cliques which favors similar classes at neighboring pixels:

$$\bar{V}_2(\bar{\omega}_C) = \gamma\delta(\bar{\omega}_C) \tag{27}$$

$$\text{with } \delta(\bar{\omega}_C) = \delta(\bar{\omega}_s, \bar{\omega}_r) = \begin{cases} 0 & \text{if } \bar{\omega}_s = \bar{\omega}_r, \\ 1 & \text{otherwise.} \end{cases} \tag{28}$$

(29)

$\mathscr{C}^i$ is the set of cliques and $\mathscr{S}^i$ is the set of sites on the grid i. $V_1^i(\omega_{s^i}^i, \mathscr{F})$ (resp. $V_2^i(\omega_C^i)$) denotes the first (resp. second) order potentials at level i, which are derived by simple computation from the potentials on the finest grid (for more details see Ref. [12]:

$$V_1^i(\omega_{s^i}^i, \mathscr{F}) = \sum_{s\in b_{si}^i} \left(-\ln(\sqrt{2\pi}\sigma_{\omega_s}) - \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right), \tag{30}$$

$$V_2^i(\omega_C^i) = \beta q^i \delta(\omega_C^i), \tag{31}$$

(32)

where $b_{s^i}^i$ denotes the block of pixels which corresponds to the site s^i in the pyramid. q^i denotes the number of cliques siting astride two neighboring blocks at scale $\mathscr{B}^i$. For example, considering 2×2 blocks and a first-order neighborhood system, we simply get $q^i = 2^i$.

Now, let us discuss the parameter estimation of the hierarchical model. It is clear from the above equations that we have the following logarithmic likelihood function:

$$\sum_{i=0}^{M} \sum_{s^i\in\mathscr{S}^i} \sum_{s\in b_{si}^i} \left(-\ln(\sqrt{2\pi}\sigma_{\omega_s}) - \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right)$$
$$- \beta \underbrace{\sum_{i=0}^{M} q^i \sum_{C^i\in\mathscr{C}^i} \delta(\hat{\omega}_{C^i})}_{N^{ih}(\hat{\omega})} - \gamma \underbrace{\sum_{C\in\bar{\mathscr{C}}_3} \delta(\hat{\omega}_C)}_{\bar{N}^{ih}(\hat{\omega})} - \ln(Z(\beta,\gamma)), \tag{33}$$

$N^{ih}(\hat{\omega})$ denotes the number of inhomogeneous cliques siting at the same scale and $\bar{N}^{ih}(\hat{\omega})$ denotes the number of inhomogeneous cliques siting astride two neighboring levels in the pyramid. First, let us consider the first term:

$$\sum_{i=0}^{M} \sum_{s^i\in\mathscr{S}^i} \sum_{s\in b_{s^i}^i} \left(-\ln(\sqrt{2\pi}\sigma_{\omega_s}) - \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right)$$
$$= \sum_{\lambda\in\Lambda} \sum_{i=0}^{M} \sum_{s^i\in\mathscr{S}_\lambda^i} \sum_{s\in b_{s^i}^i} \left(-\ln(\sqrt{2\pi}\sigma_\lambda) - \frac{(f_s - \mu_\lambda)^2}{2\sigma_\lambda^2} \right), \tag{34}$$

where $\mathscr{S}_\lambda^i$ is the set of sites at level i where $\hat{\omega}_{s^i} = \lambda$. Derivating with respect to μ_λ and σ_λ, we get

$$\forall\lambda\in\Lambda: \quad \mu_\lambda = \frac{1}{\sum_{i=0}^{M} |\mathscr{S}_\lambda^i|} \sum_{i=0}^{M} \sum_{s^i\in\mathscr{S}_\lambda^i} \sum_{s\in b_{s^i}^i} f_s,$$
$$\sigma_\lambda^2 = \frac{1}{\sum_{i=0}^{M} |\mathscr{S}_\lambda^i|} \sum_{i=0}^{M} \sum_{s^i\in\mathscr{S}_\lambda^i} \sum_{s\in b_{s^i}^i} (f_s - \mu_\lambda)^2. \tag{35}$$

Note that a gray-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 is m scales where $\hat{\omega}$ 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 probable that s belongs to class λ. Hence, its gray-level value f_s better characterizes the class λ. We note, however, that in practice, we only use the finest level because it reduces computing time and gives estimates quite close to the ones obtained by Eq. (35).

The derivative of the logarithmic likelihood function with respect to β and γ is given by

$$\frac{\partial}{\partial\beta}(-\beta N^{ih}(\hat{\omega}) - \ln(Z(\beta,\gamma)))$$
$$= -N^{ih}(\hat{\omega}) - \frac{\partial}{\partial\beta}\ln(Z(\beta,\gamma)), \tag{36}$$

$$\frac{\partial}{\partial\gamma}(-\gamma\bar{N}^{ih}(\hat{\omega}) - \ln(Z(\beta,\gamma)))$$
$$= -\bar{N}^{ih}(\hat{\omega}) - \frac{\partial}{\partial\gamma}\ln(Z(\beta,\gamma)). \tag{37}$$

From which, we get

$$N^{ih}(\hat{\omega}) = \frac{\sum_{\omega\in\Omega} N^{ih}(\omega)\exp(-\beta N^{ih}(\omega) - \gamma \bar{N}^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega) - \gamma \bar{N}^{ih}(\omega))}, \quad (38)$$

$$\bar{N}^{ih}(\hat{\omega}) = \frac{\sum_{\omega\in\Omega} \bar{N}^{ih}(\omega)\exp(-\beta N^{ih}(\omega) - \gamma \bar{N}^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega) - \gamma \bar{N}^{ih}(\omega))}. \quad (39)$$

The solution of the above equations, as in the monogrid case, can be obtained using Algorithm 3.1 with some modifications as presented below.

Algorithm 3.3. (Hierarchical hyperparameter estimation)

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

② *Using Metropolis algorithm at a fixed temperature* T, *generate a new labeling* η *sampling from*

$$P(\mathscr{X} = \omega) = \frac{\exp(-\hat{\beta}^k/T\sum_{i=0}^{M}\sum_{\{s,r\}\in\mathscr{C}^i}\delta(\omega_s, \omega_r) + \hat{\gamma}^k/T\sum_{\{s,r\}\in\bar{\mathscr{C}}}\delta(\omega_s, \omega_r))}{Z(\hat{\beta}^k, \hat{\gamma}^k)}. \quad (40)$$

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

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

Algorithm 3.2 can also be applied to the hierarchical model with trivial modifications. Hereafter, we give the algorithm used for the simulations:

Algorithm 3.4. (Unsupervised hierarchical segmentation)

① *Given an image* $\mathscr{F}$, *initialize* β, γ; μ_λ *and* σ_λ *for each* $\lambda \in \Lambda$.

② (Estimation) *Using Algorithm* 2.1 (*ASA*), *get an estimate* $\hat{\Theta}$ *of the parameters.*

③ (Segmentation) *Given the parameters* $\hat{\Theta}$, *do an ordinary supervised segmentation to get the MAP estimate of the label field given* $\mathscr{F}$ *and* $\hat{\Theta}$.

We remark, that in Step ②, the Gaussian parameters were computed considering only the finest level and not the entire pyramid (cf. Eq. (35)).

4. Implementation on a connection machine CM200

The Connection Machine [19, 20] is a data parallel (*single instruction multiple data* — SIMD) computing system associating one processor with each pixel. This computing style is well adapted to early vision problems where a large mass of data need to be processed. On the other hand, algorithms related to MRF models usually require the same local computations on a small neighborhood of each pixel.

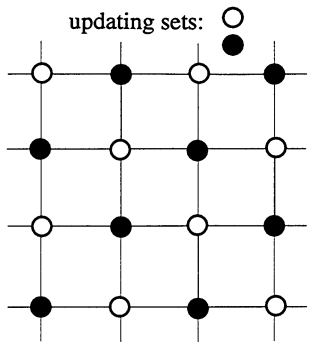

Fig. 2. Coding sets in the case of a first order monogrid MRF.

An important feature of the Connection Machine is the *virtual processor* facility. This means that a program can assume to use any appropriate number of processors (*virtual processors*) and the machine will map it onto *physical processors*. The *virtual processor ratio* (*VPR*) indicates how many times each physical processor must perform a task in order to simulate the appropriate number of virtual processors. Indeed, the greater the VPR, the more time consuming the computation.

As MRF models require computation over a small neighborhood of each pixel, fast interprocessor communication capability is especially important. The Connection Machine offers an efficient nearest-neighbor communication called NEWS ("North, East, West, South").

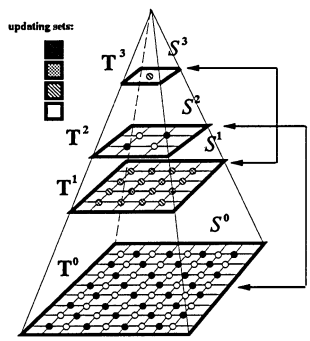

Fig. 3. Relaxation scheme on the pyramid. The levels connected by arrows are updated at the same time.

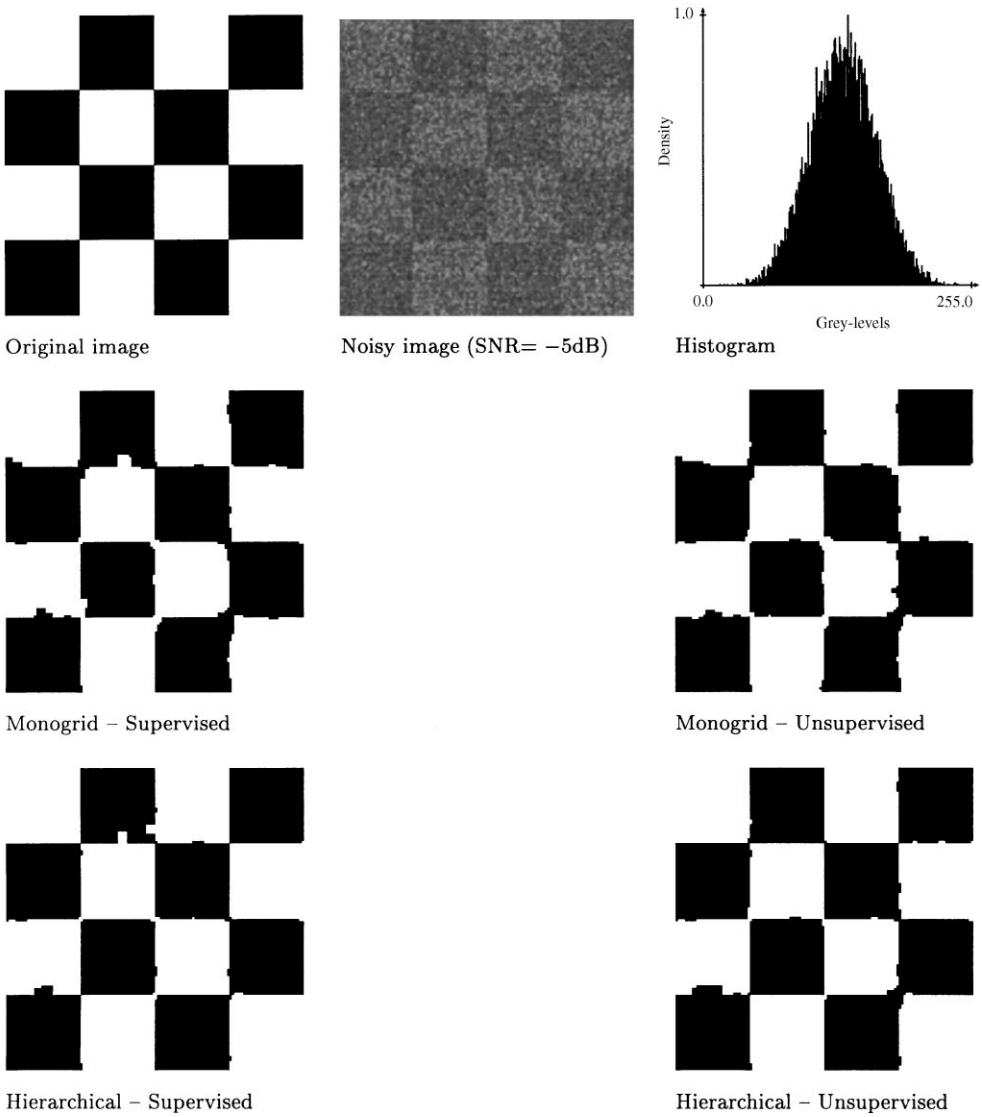

Fig. 4. Supervised and unsupervised segmentation results on the "checkerboard" image with two classes.

Thanks to a specialized hardware support, NEWS grids of any dimension can be handled with great speed. If we are working with the monogrid model, we use this type of communication. If we have no such regularity in the model, we have to use the general communication via the router. In this case, each processor can send data to or receive data from any other processor. Of course, the time required to deliver the message is much larger than in the previous case. For the hierarchical model, we must use this type of communications for the inter-level interactions. This is the main reason of the high computing time needed for the optimization of the associated energy function.

We can easily parallelize the algorithms described in this paper using the coding technique proposed by Besag in Ref. [3]. It consists of constructing *coding sets* such that pixels belonging to the same set are conditionally independent, given the data of all the other sets. Thus, pixels belonging to the same coding set can be updated at the same time. We show the coding sets of the monogrid model in Fig. 2 and those of the hierarchical model in Fig. 3.

5. Experimental results

We have tested the proposed monogrid and hierarchical unsupervised algorithms on noisy synthetic and real images. The algorithms were implemented on a Connection Machine CM200 [19, 20] We have

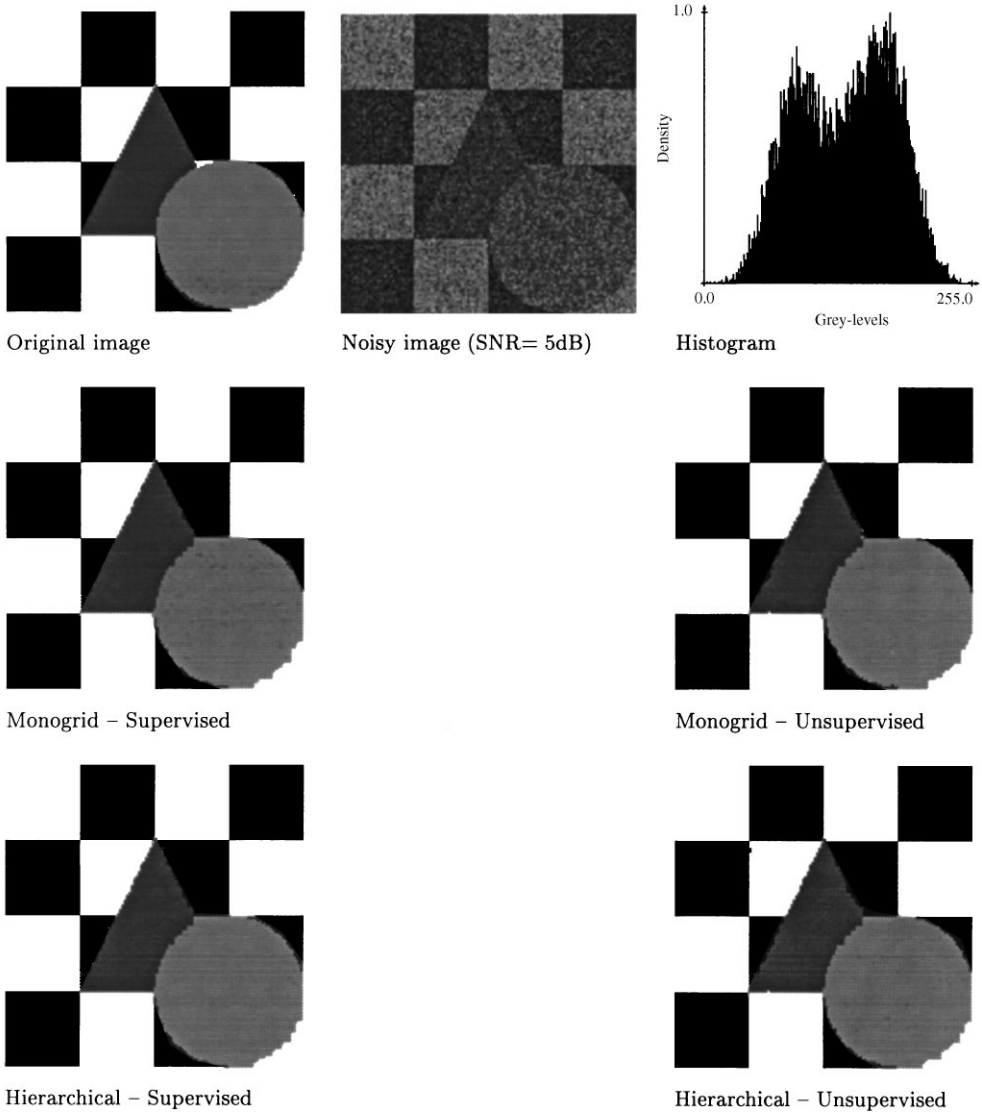

Fig. 5. Supervised and unsupervised segmentation results on the "triangle" image with four classes.

compared the obtained parameters and segmentation results to the supervised results already presented in Ref. [12]. In general, the quality of unsupervised results are as good, or sometimes slightly better, than the results of supervised segmentation. We observed, however, that the unsupervised algorithm is more sensitive to noise than the supervised one. This is due to the initial conditions, in particular, the initialization of the mean and the variance of the classes (the initialization of β and γ are not crucial). For example, in the case of the "triangle" image (see Fig. 5) with SNR $= 3\,d$B one class has been lost. But with SNR $= 5\,d$B, the result is as good as for the supervised algorithm.

Table 1
Comparison of supervised and unsupervised segmentation results. (Number of misclassified pixels)

Model	Image	Supervised	Unsupervised
Monogrid	Checkerboard	260 (1.59%)	213 (1.41%)
	Triangle	112 (0.68%)	103 (0.63%)
Hierarchical	Checkerboard	115 (0.7%)	147 (0.9%)
	Triangle	104 (0.63%)	111 (0.68%)

Before evaluating the results, let us explain some important points of the implementation. The only parameter which has to be defined by the user is the number

Table 2
Parameters of the "checkerboard" image

Monogrid model				Hierarchical model			
	Unsupervised				Unsupervised		
Parameter	Initial	Final	Supervised	Parameter	Initial	Final	Supervised
μ_0	123.5	117.3	119.2	μ_0	123.5	126.7	119.2
σ_0^2	256.0	680.0	659.5	σ_0^2	256.0	903.4	659.5
μ_1	170.0	151.5	149.4	μ_1	170.0	151.5	149.4
σ_1^2	169.0	668.2	691.4	σ_1^2	169.0	689.3	691.4
β	0.7	0.7	0.9	β	0.7	0.7	0.7
				γ	0.1	0.1	0.3

Table 3
Computer time of the "checkerboard" image

Model	VPR	Total CPU time (s)	Estimation (s)	Segmentation (s)
Monogrid	2	142.73	133.57	9.16
Hierarchical	4	1551.93	1042.46	446.52

Table 5
Computer time of the "triangle" image

Model	VPR	Total CPU time (s)	Estimation (s)	Segmentation (s)
Monogrid	2	249.75	237.00	12.75
Hierarchical	4	1762.23	1232.82	529.41

of classes. All the other parameters are automatically estimated from the data. Essentially, we have followed Algorithm 3.2. First, the initial values of the mean and variance have been estimated: we have used a method proposed by Postaire and Vasseur [9] which consists of the geometrical analysis of the histogram, regarded as a Gaussian mixture, in order to determine its modes. For the hyperparameters, we have chosen as initial values $\beta = 0.7$ and $\gamma = 0.1$. Experiments show that these initial values are not vital, practically any value between 0.5 and 1 is good for β and a value close to zero is good for γ.

In the next step (Step ② of Algorithm 3.2), we use the ICE algorithm (see Algorithm 2.2) to iteratively reestimate the parameters. We have chosen ICM to generate labelings because of its rapidity: Given the parameters

Table 4
Parameters of the "triangle" image

Monogrid model				Hierarchical model			
	Unsupervised				Unsupervised		
Parameter	Initial	Final	Supervised	Parameter	Initial	Final	Supervised
μ_0	83.5	84.3	85.48	μ_0	83.5	84.3	85.48
σ_0^2	256.0	480.5	446.60	σ_0^2	256.0	483.9	446.60
μ_1	100.0	117.3	115.60	μ_1	100.0	115.5	115.60
σ_1^2	169.0	416.3	533.97	σ_1^2	169.0	444.6	533.97
μ_2	152.5	148.1	146.11	μ_2	152.5	146.7	146.11
σ_2^2	676.0	457.8	540.32	σ_2^2	676.0	502.1	540.32
μ_3	181.5	178.5	178.01	μ_3	181.5	177.9	178.01
σ_3^2	100.0	490.9	504.34	σ_3^2	100.0	500.0	504.34
β	0.7	1.0	1.0	β	0.7	1.0	0.7
				γ	0.1	0.1	0.1

Table 6
Parameters of the "holland" SPOT image

Parameter	Unsupervised		Supervised
	Initial	Final	
μ_0	51.5	53.1	54.6
σ_0^2	36.0	10.3	93.1
μ_1	60.0	77.2	73.5
σ_1^2	49.0	64.3	4.1
μ_2	70.5	89.6	82.5
σ_2^2	49.0	30.7	35.5
μ_3	80.5	102.5	93.8
σ_3^2	64.0	35.7	93.7
μ_4	97.5	116.2	100.5
σ_4^2	441.0	27.6	308.8
μ_5	122.5	127.2	122.8
σ_5^2	484.0	18.9	8.9
μ_6	136.0	138.6	129.9
σ_6^2	1.0	20.2	37.4
μ_7	152.5	152.7	146.6
σ_7^2	625.0	18.0	15.3
μ_8	169.0	162.4	159.9
σ_8^2	1.0	7.4	31.3
μ_9	181.5	174.2	182.3
σ_9^2	25.0	54.1	73.1
β	0.7	1.3	1.0

$\hat{\Theta}^n$, the ICM is used to maximize the *a posteriori* probability of the label field ω. Suppose that ICM converges in N iterations (N is typically less than 10) given N realizations of ω. Using these labelings, we have to compute N ML estimates of Θ (see Algorithm 2.2 for more details).

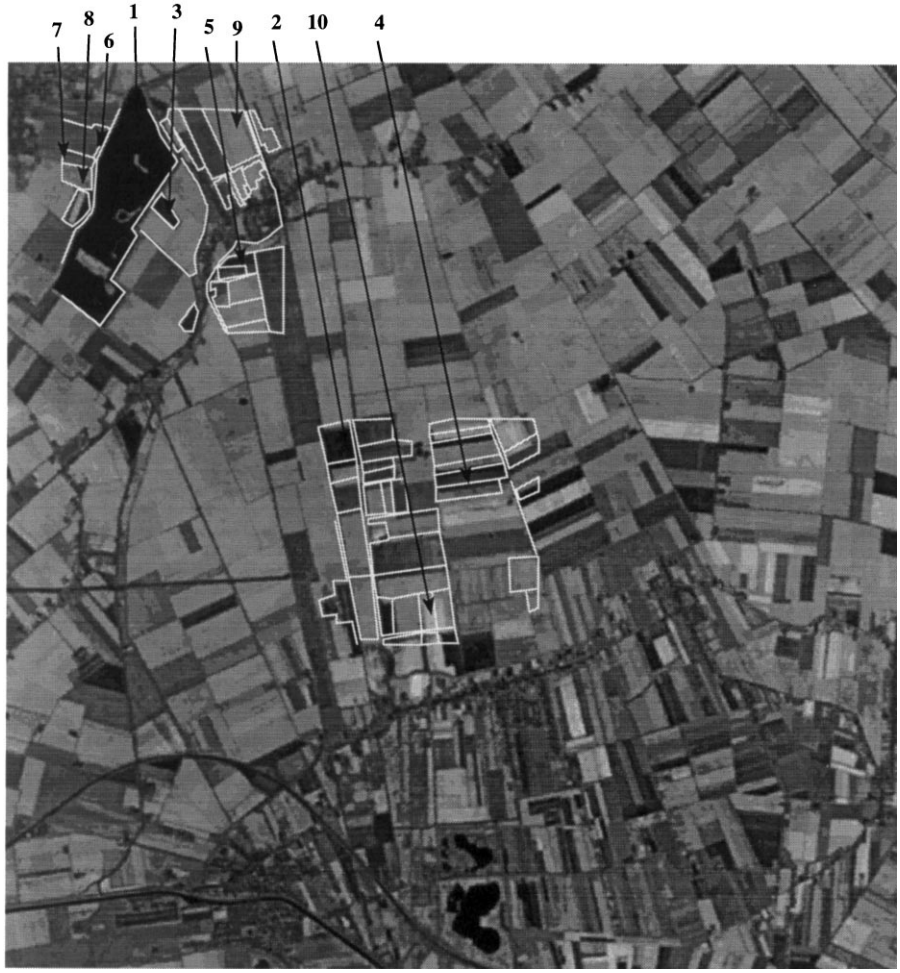

Fig. 6. Training areas on the "holland" SPOT image with ground truth data.

Table 7
Computer time of the "holland" SPOT image

Model	VPR	Total CPU time (s)	Estimation (s)	Segmentation (s)
Monogrid	32	3576.58	3270.78	305.81

Fig. 7. Supervised segmentation result with 10 classes.

For the hierarchical model, however, we have used ASA (cf. Algorithm 3.4), because the ICE algorithm would be too slow with such a model: Using ICM, we maximize the *a posteriori* probability of ω, given the parameter estimates $\hat{\Theta}''$. Then, the ML estimate is computed based on the obtained labeling. Another modification is that the Gaussian parameters were computed considering *only the finest level* and not the entire pyramid as explained in Section 3.2. This is because the variances obtained with the original algorithm were to large. This modification also reduces the computing time.

Once the sequence $\hat{\Theta}''$ becomes steady, the estimation step is completed and one proceeds to the segmentation (with known parameters) using the Gibbs sampler, for instance.

The algorithms were tested on the "checkerboard" (Fig. 4), "triangle" (Fig. 5) and "holland" SPOT (Figs. 6–8) images. For the synthetic images, we also give the histogram, since the initial estimates are based on it. In Table 2, Table 4 and Table 6, we compare the parameters obtained by the unsupervised algorithm to the ones used for the supervised segmentation. We remark that the parameters of the supervised algorithm are not

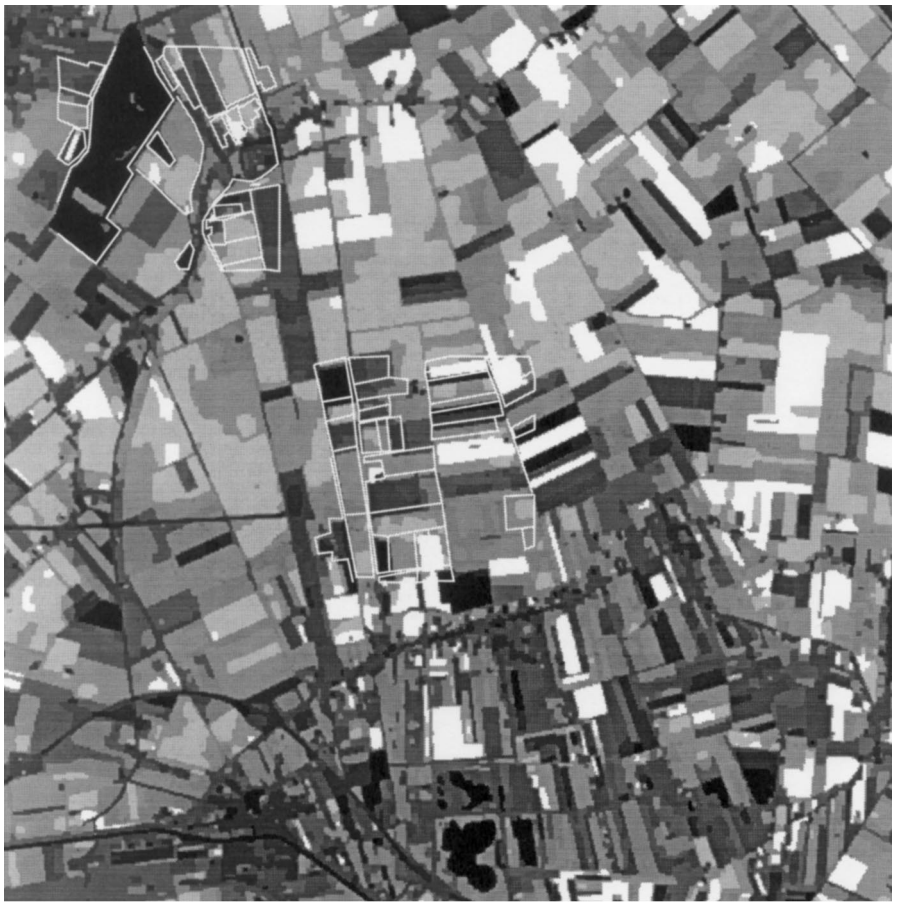

Fig. 8. Unsupervised segmentation result with 10 classes.

necessarily correct. They have been computed on training sets selected by an expert (cf. Fig. 6). In Table 3, Table 5 and Table 7, we give the computer time of the estimation and segmentation steps. As we can see, the estimation requires much more time than the segmentation. The hyperparameter estimation requires the largest part of the computer time since it consists of generating new labelings by Metropolis algorithm in Step ② of Algorithm 3.1.

Table 1 provides an objective comparison of supervised and unsupervised segmentation results based on the number of misclassified pixels. The obtained results are practically the same for supervised and unsupervised segmentation.

6. Conclusion

Developing a completely data-driven algorithm for image classification is an extremely difficult problem. We have presented some iterative unsupervised parallel segmentation algorithms for both monogrid and hierarchical Markovian models. The first results are encouraging but unsupervised algorithms require much more computing time due to the hyperparameter estimation. In the current implementation, they are computed using Metropolis algorithm, which is very time consuming. Mean-field approximation would probably result in a faster convergence [21, 22]. Another important point is the initialization of the Gaussian parameters for each class. We have noted that unsupervised algorithms are more sensitive to noise than supervised ones. This sensitivity is due to bad initial conditions in the case of noisy images.

In summary, the presented unsupervised algorithms provide results comparable to those obtained by supervised segmentations, but they require much more computing time and they are slightly more sensitive to noise. The main advantage is, of course, that unsupervised methods are completely data-driven. The only input parameter is the number of classes.

References

[1] Y. Bard, Nonlinear Parameter Estimation, Academic Press, New York, 1974.

[2] B. Chalmond, Image restoration using an estimated markov model, Signal Process. 15 (1988) 115–129.

[3] J. Besag, On the statistical analysis of dirty pictures, J. Roy. Statist. Soc. B. 62 (1986) 259–302.

[4] H. Caillol, A. Hillion, W. Pieczynski, Fuzzy random fields and unsupervised image segmentation, IEEE Geosci. Remote Sensing 31 (1993) 801–810.

[5] P. Masson, W. Pieczynski, SEM algorithm and unsupervised statistical segmentation of satellite images, IEEE Geosci. Remote Sensing 31 1993 618–633.

[6] W. Pieczynski, Statistical image segmentation, Machine Graphics and Vision, GKPO'92, Naleczow, Poland. May 1992 pp. 261–268.

[7] K. Fukunaga, T. Flick, Estimation of the parameters of a Gaussian mixture using the method of moments, IEEE Trans. Pattern Anal. Mach. Intell. 5, 1983 410–416.

[8] H. Derin, Estimation components of univariate Gaussian mixtures using prony's method, IEEE Trans. Pattern Anal. Mach. Intelli. 9 1987 142–148.

[9] J.G. Postaire, C.P.A. Vasseur, An approximate solution to normal mixture identification with application to unsupervised pattern classification, IEEE Trans. Pattern Anal. and Mach. Intell. 3 1981 163–179.

[10] Z. Kato, J. Zerubia, M. Berthod, Bayesian image classification using markov random fields, in: Maximum Entropy and Bayesian Methods, A.M.-D.G. Demoments, Kluwer Academic Publisher, Dordrecht, 1993, pp. 375–382..

[11] C. Graffigne, F. Heitz, P. Pérez, F. Prêteux, M. Sigelle, J. Zerubia, Hierarchical markov random field models applied to image analysis: a review. IEEE Trans. on Inform. Theory, 1997, submitted.

[12] Z. Kato, M. Berthod, J. Zerubia, A hierarchical Markov random field model and multi-temperature annealing for parallel image classification, Comput. Vision Graphics Image Process. Graphical Models Image Process. 58 (1996) 18–37.

[13] D. Geman, Bayesian image analysis by adaptive annealing, in Proc. IGARSS'*85*, pp. 269–277. Amherst, USA October 1985.

[14] S. Lakshmanan, H. Derin, Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1989) 799–813.

[15] O. Allagnat, J.M. Boucher, D.C. He, W. Pieczynski, Hidden Markov fields and unsupervised segmentation of images, Proc. ICPR'92, 1992.

[16] B. Braathen, W. Pieczynski, P. Masson, Global and local methods of unsupervised Bayesian segmentation of images, Mach. Graphics Vision. 2(1) (1993) 39–52.

[17] Z. Kato, M. Berthod, J. Zerubia, W. Pieczynski, Unsupervised adaptive image segmentation, Proc. ICASSP'95. Detroit, U.S.A., May, 1995.

[18] R. J. Baxter, Exactly Solved Models in Statistical Mechanics. Academic Press, New York, 1990.

[19] W. D. Hillis, The Connection Machine, MIT Press, New York, 1985.

[20] Thinking Machines Corporation, Cambridge, Massachusetts, Connection Machine Technical Summary, version 5.1 ed., 1989.

[21] D.A. Langan, K.J. Molnar, J.W. Modestino, J. Zhang, Use of the mean-field approximation in an EM-based approach to unsupervised stochastic model-based image segmentation, Proc. ICASSP'92, San Francisco, 1992, pp. III–57–III–60.

[22] J. Zerubia, R. Chellappa, Mean field annealing using compound Gauss–Markov random fields for edge detection and image estimation, IEEE Trans. Neural Networks 8 (1993) 703–709.

About the Author—ZOLTAN KATO received the M.S. degree in computer science from the Jozsef Attila University, Hungary in 1990 and the Ph.D. degree in computer science from the University of Nice, France in 1994 doing his research work at INRIA-Sophia Antipolis, France. He joined the Computer Science Department of the Hong Kong University of Science and Technology as a visiting research associate in 1996. He is currently a visiting research associate at CWI, The Netherlands granted by an ERCIM postdoctoral fellowship. His research interest includes image segmentation, color, motion, content based indexing and retrieval, statistical image models, Markov Random Fields, combinatorial optimization, parameter estimation.

About the Author—JOSIANE ZERUBIA is a permanent research scientist at INRIA since 1989. She is director of research since July 1995. She was head of a remote sensing laboratory (PASTIS, INRIA Sophia-Antipolis) from mid-1995 to 1997. Since January 1998, she is in charge of a new research group working on remote sensing (ARIANA, INRIA-CNRS-University of Nice).

Before, she was with the Signal and Image Processing Institute of the University of Southern California (USC) in Los-Angeles as a post-doc. She also worked as a researcher for the LASSY (University of Nice and CNRS) from 84 to 88 and in the Research Lab. of Hewlett Packard in France and in Palo-Alto (CA) from 82 to 84.

She has got an Electrical Engineer degree from ENSIEG, Grenoble, France in 81; a Doctor Engineer degree in 86; a Ph.D. in 88 and an "Habilitation" in 94.

She is member of IEEE since 1988 and of the NY academy of sciences since 1996. She is part of the IEEE IMDSP Technical Committee (SP Society) since 1997 and associate editor of IEEE Trans. on IP since 1998.

Her current interest is image processing (image restoration, image segmentation or classification, line detection, perceptual grouping, stereovision, super-resolution) using probabilistic models or neural networks. She also works on parameter estimation and optimization techniques.

About the Author—MARC BERTHOD graduated from Ecole Polytechnique in 1969. He worked first on pattern recognition, defending a third cycle thesis on on-line character recognition in 1975, and a Ph.D. thesis on relaxation labeling in 1980. From 1982 to 1995, he led a research group working on computer vision and more specifically symbolic image analysis at INRIA Sophia-Antipolis. His research interests are 3-D reconstruction from satellite data, symbolic interpretation in remote sensing, and Markov modelization and associated optimization methods, mostly applied to satellite image interpretation. He took in charge INRIA's Sophia-Antipolis research unit in July 1996.