

Discrete Markov Image Modeling and Inference on the Quadtree

Jean-Marc Laferté, Patrick Pérez, and Fabrice Heitz

Abstract—Noncasual Markov (or energy-based) models are widely used in early vision applications for the representation of images in high-dimensional inverse problems. Due to their noncausal nature, these models generally lead to iterative inference algorithms that are computationally demanding. In this paper, we consider a special class of nonlinear Markov models which allow to circumvent this drawback. These models are defined as discrete Markov random fields (MRF) attached to the nodes of a quadtree. The quadtree induces causality properties which enable the design of exact, noniterative inference algorithms, similar to those used in the context of Markov chain models. We first introduce an extension of the Viterbi algorithm which enables exact maximum a posteriori (MAP) estimation on the quadtree. Two other algorithms, related to the MPM criterion and to Bouman and Shapiro's sequential-MAP (SMAP) estimator are derived on the same hierarchical structure. The estimation of the model hyper-parameters is also addressed. Two expectation-maximization (EM)-type algorithms, allowing unsupervised inference with these models are defined. The practical relevance of the different models and inference algorithms is investigated in the context of image classification problem, on both synthetic and natural images.

Index Terms—Discrete Markov random field (MRF), expectation-maximization (EM), hierarchical modeling, maximum a posteriori (MAP), modes of posterior marginal (MPM), noniterative inference, quadtree independence graph, sequential-MAP (SMAP), supervised and unsupervised classification.

I. INTRODUCTION

NONCASUAL Markov random field (MRF) models have been extensively used for modeling spatial interactions between various attributes of an image [19]. MRF models have thus become a major ingredient for most Bayesian approaches in early vision. For most noncausal representations, the graph associated to the Markov model [48] is the rectangular lattice equipped with the nearest (or second nearest) neighborhood system. From an estimation point of view, this kind of graph results in iterative procedures which propagate the available information back and forth, so that each hidden variable is eventually estimated given all the data. These algorithms,

similar, in the case of continuous-valued variables, to the resolution of discretized partial differential equations, are computationally demanding.

Beside these standard lattice-based models, other Markovian interaction structures have been considered, associated to causality properties in the image plane or, more recently, to causality in scale. Among others, dyadic trees and quadtrees have been proposed as attractive candidates for modeling monodimensional [1], [7], [14], [18], [38] and bidimensional [4], [15], [16], [34]–[36], signals. With these hierarchical probabilistic models, the standard spatial prior captured by lattice-based models is replaced by a fractal-type prior based on scale-to-scale interactions. Both experimental and theoretical considerations indicate that such a prior is a good alternative to spatial interaction priors [10], [17], [35]. In addition, tree-based models are appealing from an algorithmic point of view for they enable the design of *noniterative* inference procedures similar to those used for discrete and continuous Markov chain models.

Note that the causality property, which is at the midst of the noniterative inference capabilities of these models, has also been investigated for long from a purely spatial point of view. The idea is then to define causality with respect to some spatial ordering of variables (e.g., lexicographical ordering of the image sites). Various instances of this approach have been reported such as the Pickard random field [12], [40], the mutually compatible MRF's [22], or the more standard Markov chain image model [21]. Pickard random fields and Markov chain models are however known to represent only a limited class of spatial statistics and generally yield directional artifacts in the image plane. On the other hand, accurate causal approximations of noncausal MRF's can be obtained [22], [37].

We consider here discrete-valued (nonlinear) causal models defined on the quadtree, as an appealing alternative to standard noncausal lattice-based models. The analogy of quadtree-based models with chain-based representations has already been thoroughly investigated in the continuous Gaussian case, where Kalman filtering is a key tool [1], [14], [16], [30], [34]. The discrete (nonlinear) case has received far less attention, apart from the work by Bouman and his colleagues [4], [46], in which an approximate noniterative inference algorithm is proposed. In this paper we introduce several algorithms for both supervised and unsupervised inference with discrete hidden Markov random field models supported by trees. More precisely, we present:

- 1) Viterbi-like algorithm [18] computing, in a noniterative way, the exact maximum *a posteriori* (MAP) estimate on the quadtree;

Manuscript received July 21, 1998; revised July 23, 1999. This work was supported in part by MENESR through a student grant and by the GdR/PRC Isis. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Josiane B. Zerubia.

J.-M. Laferté is with IRISA/University of Rennes, 35042 Rennes Cedex, France (e-mail: jlaferte@irisa.fr).

P. Pérez is with IRISA/INRIA, 35042 Rennes Cedex, France (e-mail: perez@irisa.fr).

F. Heitz is with Ecole Nationale Supérieure de Physique, Laboratoire des Sciences de l'Image, de l'Informatique et de la Teledetection, Strasbourg I University, 67400 Illkirch, France (e-mail: Fabrice.Heitz@ensps.u-strasbg.fr).

Publisher Item Identifier S 1057-7149(00)01801-7.

- 2) noniterative procedure that provides the exact computation of point-wise and pair-wise posterior marginals, with the modes of posterior marginal (MPM) estimate as a byproduct;
- 3) generalization of the noniterative algorithm introduced by Bouman and Shapiro [4] for a statistical inference according to the so-called “sequential-MAP” (SMAP) criterion.

As for parameter estimation, we introduce two original EM-type algorithms for tree-based models. The first one corresponds to the exact deterministic application of plain EM iterations. The second one is a stochastic variant of EM, based on noniterative sampling from the posterior distribution, which allows to circumvent the critical problem of initialization.

The paper is organized as follows. After a brief review of hierarchical MRF’s in Section II, we introduce the notations in Section III and specify the statistical properties of hidden MRF’s on the (quad)tree. Three noniterative inference algorithms, associated to the MAP, MPM, and SMAP criteria, are derived in Section IV. These algorithms are experimentally compared in a standard supervised classification problem. Section V is devoted to the parameter estimation problem: Nonsupervised EM-type procedures are defined on the quadtree, for the estimation of both prior model parameters and data likelihood parameters. Experimental results include comparisons with standard nonhierarchical approaches.

II. HIERARCHICAL MARKOV IMAGE MODELS

Although the concept of hierarchical processing has almost always been present in early vision, a satisfactory treatment of this subject in the framework of statistical models dates from the late eighties. Since then, it has received an increasing attention from both the computer vision and signal processing communities.

The motivations for hierarchical models are threefold.

- Need in statistical image modeling for algorithms that are able, like multigrid techniques in numerical analysis, to provide fast computations and low sensitivity to initial conditions and distracting local minima.
- Need for statistical models that are able to capture the intrinsic hierarchical nature of data (fractal images and signals, multiscale phenomena such as turbulence, etc.).
- Need for efficient tools able to process multimodal data that, in many applications, come in ever increasing volumes, with a variety of resolution and spectral domains (e.g., multispectral and multiresolution satellite images in remote sensing).

Three main types of approaches have been investigated:

- approaches related to the renormalization group theory from statistical physics, that derive reduced probabilistic models from a given original spatial model and fine-to-coarse deterministic or stochastic transformations [13], [20], [30], [32], [42];
- multigrid-like approaches in which the inference is conducted within decreasingly constrained subsets of configurations [24];

- modeling approaches that aim at defining right away hierarchical models on trees [4], [15], [31], [34] or on other hierarchical graphs [8], [29].

The models we investigate here are of the third kind, since they are specified on a hierarchical graph. They are thus manipulated as a whole, leading to a unique statistical inference problem instead of a sequence of multiresolution problems loosely related. Apart from its simplicity, the tree yields in-scale causality properties allowing fast noniterative inference procedures. Chou *et al.* [7] have thus defined Gaussian models on dyadic trees for monodimensional signal modeling. They derive noniterative estimation procedures on the dyadic tree, corresponding to Kalman-type filtering through scales. The theoretical study of in-scale causal autoregressive models has been conducted on infinite p -adic trees [1]. The Gaussian representation has been extended by Luetgten *et al.* [34]–[36] on the quadtree; it has been applied to various tasks in early vision such as the estimation of optic flow [34], the recovery of sea-surface height [16], or the reconstruction of surfaces [15].

Whereas the previous models are Gaussian (hence, continuous and linear), Bouman and Shapiro [4] worked at designing discrete (nonlinear) models on the same hierarchical structure. They also introduced a new Bayesian estimation criterion, the “sequential MAP” (SMAP) criterion, which is better suited to hierarchical modeling than the standard MAP criterion (the SMAP is described in Section IV-C). They derived a noniterative inference procedure on the quadtree that computes an approximate SMAP estimate. This procedure requires two passes on the tree, and is applied to image classification, segmentation, and inspection problems [4], [46].

It should be mentioned that the quadtree structure induces nonstationarity in space (the distribution at leaves is not shift-invariant since the correlation between two variables depends on the “distance” to their common ancestor in the tree). This may result in block artifacts in the final estimates. The blocky aspect of the estimates has been reported by all authors [4], [7], [15], [16], [31], [34]. Several techniques have been proposed to alleviate such undesired effects (e.g., *a posteriori* smoothing [34], definition of tree structures with overlapping data leaves [25]). We do not deal with this issue here, although these techniques could probably be extended to the models and algorithms described in this paper.

Another way to circumvent block effects consists in using hierarchical graph structures that are more complex than mere trees. Unfortunately, the practical advantages of the tree structure are then partly or completely lost. Bouman and Shapiro [4] add for instance *inter-level edges* to the original quadtree to get a more interleaved structure that avoids, at least partially, block artifacts. Exact inference on this new graph structure leads to iterative algorithms, but an approximate noniterative method is proposed by the authors. Kato *et al.* [29] consider a more complex graph in which the original quadtree is combined with a spatial lattice neighborhood at each level. This results in an interleaved model whose manipulation is iterative, with a complexity per pixel even higher than the one exhibited by standard spatial models. The inference is conducted using a modified annealing procedure where temperature is kept high at the coarse levels of the structure. This approach provides excellent

classification results but is computationally demanding: one of the advantages of hierarchical approaches (i.e., reduction of the computational complexity) is lost in this case [8], [29].

In Section IV, we derive three noniterative algorithms that provide fast MAP, SMAP, and MPM estimates on the quadtree. These estimates are exact in the case of MAP and MPM and their quality should be considered as satisfactory in many applications. In the next section, we first recall the main properties of the discrete Markov model on the quadtree and introduce some necessary notations.

III. MARKOV MODELS ON THE QUAD-TREE

A. Problem Statement and Notations

We consider a standard inverse problem in which one attempts to estimate the “best” realization of some hidden variable set x given another set of observed variables y which is somehow related to the former one. As usual in statistical approaches, x and y are viewed as occurrences of some random vectors X and Y whose Markovian independencies can be represented by an *independence graph* [48]: the components of vectors X and Y (i.e., the random variables) are attached to a set of nodes, and any two nodes are *not neighbors* if they support two random variables that are independent given all others. Equivalently, the joint distribution (X, Y) factorizes as a product of “local” functions in such a way that two nodes are neighbors if they support two random variables that simultaneously appear in a same factor of the decomposition. As for X itself (i.e., the prior model), this graph is often the regular rectangular lattice that fits the grid of pixels, and that is equipped with a four- or eight-neighborhood system. The structure of the joint model is then usually obtained by attaching one observation node (corresponding to one component of Y) to each node of the latter prior graph.

It turns out that inference of x given $Y = y$ is all the more computationally demanding since the independence graph is complex. In most cases, iterative algorithms are required, and their speed of convergence decreases as the number of cycles in the graph gets larger.

For both computational and modeling reasons which will become clear later, we consider here a particular hierarchical graph structure, namely the tree. The components of X are thus assumed to be indexed by the nodes of a *quadtree* (see Fig. 1), i.e., a tree in which each node (apart from the leaves) has four off-springs. We now introduce a few notations.

The set of nodes of the tree is denoted S , and its *root* is referred to as site r . Any node different from r has a unique *parent node* denoted s^- , where superscript “-” recalls that the resolution (or depth) decreases when going from a node to its parent. Conversely, the set of the four children of any node that is not a leaf is denoted $t^+ \triangleq \{s: t = s^-\}$. A *descendant* stemming from s is a node t such that s belongs to the unique chain that joins t to the root. The set of descendants of s , including s itself, is denoted $d(s)$. The nodes belonging to the same “generation” n from the root form the “resolution level” S^n of the tree. The coarsest level reduces to the root node: $S^0 = \{r\}$. The last

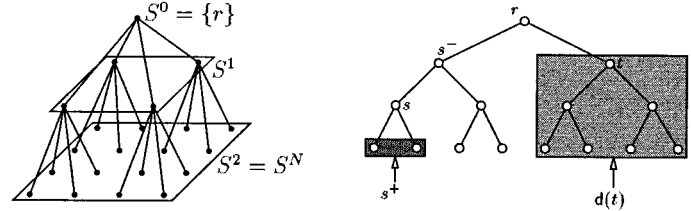


Fig. 1. Quadtree graph structure and notations on the tree.

level (the “finest” resolution) is S^N for some positive integer N . These different notations are summarized in Fig. 1 (where, for graphical convenience, the second quadtree has been represented by a dyadic tree).

B. Statistical Modeling

We now come back to random vectors X and Y which are assumed to be discrete. The components of random vector X are indexed by the nodes of S , and take their values in a discrete set Λ . In particular, one can define the restriction of X to level S^n : $X^n \triangleq (X_s)_{s \in S^n}$. The restriction to any other site subset $a \subset S$ will be denoted by $X_a \triangleq (X_s)_{s \in a}$. Similar notations stand for occurrences of X : a *configuration* is a vector $x = (x_s)_{s \in S}$ from configuration set $\Omega \triangleq \Lambda^S$, which may be partitioned as $x = (x^n)_{n=0}^N$. In *classification* problems, for instance, each random variable takes its values in a finite set of class labels $\Lambda = \{1, \dots, M\}$, where M is the total number of classes, and each $x^n = (x_s)_{s \in S^n}$ corresponds to a possible classification at resolution level n .

In the same way, the observation vector Y is assumed to be indexed by S . Data are often grey level images, each component of Y taking for instance its values within $\{0, \dots, 255\}$. In practice (especially, when the inverse problem concerns a single image), such observations are often available at the finest level S^N only. This is however not always the case, for instance in the classification of *multiresolution data* [31]. In the following we consider that measurements (possibly multidimensional, as in multispectral classification) are available at each node $s \in S$, with state space Υ . All derivations can be easily extended to other cases, when data concern only some subset of nodes.

We now make further statistical assumptions about the couple of random vectors (X, Y) .¹ The two first assumptions concern the prior model (that is X), while the third one specifies the statistical interactions between X and Y .

- Markov property over scale: the partition $(X^n)_{n=0}^N$ is a first order top-down Markov chain

$$P(x^n | x^{n-1} \dots x^0) = P(x^n | x^{n-1}), \forall n \in \{1 \dots N\}. \quad (1)$$

- The transition probabilities of this Markov chain factorize such that the components of X^n are mutually independent

¹Throughout the paper, except in ambiguous cases, we shall denote \mathbb{P} the joint distribution of (X, Y) [thus $\mathbb{P}(x, y) = \mathbb{P}(X = x, Y = y)$] as well as *any conditional or marginal distribution* arising from it: for any site subsets a, b, c , and d , $\mathbb{P}(x_a, y_b | x_c, y_d)$ stands for $\mathbb{P}(X_a = x_a, Y_b = y_b | X_c = x_c, Y_d = y_d)$.

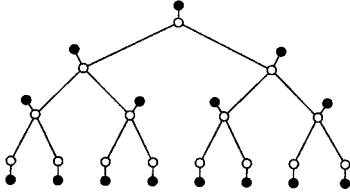


Fig. 2. Independence graph of the joint model (X, Y) , where, for graphical convenience, the quadtree is represented as a dyadic tree (white nodes). Black nodes are associated with the variable Y_s .

given X^{n-1} . Furthermore, for each node in S^n , the conditioning in X^{n-1} reduces to a dependence with respect to its parent only

$$P(x^n|x^{n-1}) = \prod_{s \in S^n} P(x_s|x_{s-}), \forall n \in \{1 \dots N\}. \quad (2)$$

- For the observation model $P(y|x)$, we assume a standard *site-wise* factorization of the form

$$P(y|x) = \prod_{s \in S} P(y_s|x_s) \quad (3)$$

which means that the components of Y are all mutually independent given X , and that for each of them, the conditioning w.r.t. X is equivalent to only conditioning w.r.t. the component of X at the same node. If the observation Y_s actually does not exist, one has to replace $P(y_s|x_s)$ by one, for any x_s .

Gathering (1)–(3), one gets the following factorization of the joint distribution

$$P(x, y) = P(x_r) \prod_{s \neq r} P(x_s|x_{s-}) \prod_{s \in S} P(y_s|x_s) \quad (4)$$

which is entirely defined by the root prior $P(x_r)$, the parent-child transition probabilities $\{P(x_s|x_{s-})\}_{s \neq r}$, and the data conditional likelihoods $\{P(y_s|x_s)\}_{s \in S}$. This factorization implies that (X, Y) is a Markov random field with respect to the quadtree (for the prior distribution) with, in addition, data sites in one-to-one correspondence with the former nodes (see Fig. 2). Note that the nodes respectively associated to X_s and to Y_s are both referred to as “ s .” This graph is the independence graph of (X, Y) , as defined in [48].

This graphical interpretation and reading of factorization (4) has one key advantage [48]: it neatly captures all conditional independencies among the components of the joint random vector $Z = (X, Y)$ in terms of graphical separation [provided that $P(x, y) > 0$ for any joint configuration $z = (x, y)$]: if a node subset a separates two other disjoint node subsets b and c in this independence graph (i.e., all chains from $s \in b$ to $t \in c$ intersect a), the random variables associated to b and c are independent given the variables associated to a . This corresponds to factorization $P(z_b, z_c|z_a) = P(z_b|z_a)P(z_c|z_a)$ and to conditioning reduction $P(z_b|z_a, z_c) = P(z_b|z_a)$. This property is particularly

powerful for independence graphs corresponding to trees. Indeed, each node s of the tree that is not a leaf separates the whole graph into at least two parts: given X_s , any set of variables within one of the parts is independent from any other set of variables within one of the other parts. This will be the most important property that makes all coming derivations possible.

IV. INFERENCE ALGORITHMS ON THE QUADTREE

We now consider the problem of inferring the “best” configuration of X from the observed data $Y = y$. The standard Bayesian formulation of this inference problem consists in minimizing the expectation of some *cost function* C , given the data

$$\hat{x} \triangleq \arg \min_{x \in \Omega} \mathbb{E}(C(X, x)|Y = y) \quad (5)$$

where C penalizes the discrepancy between the estimated configuration and the “ideal” random one.

In this section, we introduce three inference algorithms for the discrete quadtree-based model corresponding to three different cost functions. These algorithms provide respectively the exact MAP estimate, the exact MPM estimate, and an approximation of the SMAP estimate introduced by Bouman and Shapiro [4].

A. MAP Estimation

The Viterbi algorithm is a standard technique for computing the MAP estimate of hidden Markov models (HMM’s) whose prior part is a chain [18]. This algorithm is widely used, for instance, in speech recognition [14], [38]. We describe here an extension of Viterbi algorithm, which computes the exact MAP estimate of X given $Y = y$ on the tree. This extension has been independently introduced by Dawid in the context of probabilistic expert systems [11], and by Laferté *et al.* in the context of discrete image modeling [31]. The proposed algorithm is non-iterative and requires two passes on the tree.

The cost function associated to the MAP criterion is

$$\forall (x, x') \in \Omega \times \Omega, C(x, x') = 1 - \delta(x', x) \quad (6)$$

where δ is the Kronecker delta function. The corresponding Bayesian estimator is readily obtained

$$\hat{x} = \arg \max_{x \in \Omega} P(x|y) = \arg \max_{x \in \Omega} P(x, y) \quad (7)$$

and corresponds to the mode of the posterior distribution $P(x|y)$.

Using Bayes’ rule and the separation property on the tree, conditioning w.r.t. X_r yields

$$\begin{aligned} P(x, y) &= P(x_{S \setminus \{r\}}, y|x_r)P(x_r) \quad \text{Bayes' rule} \\ &= P(y_r, \{x_{a(s)}, y_{a(s)}\}_{s \in r^+}|x_r)P(x_r) \\ &\quad \text{since } S \setminus \{r\} = \bigcup_{s \in r^+} d(s) \\ &= P(y_r|x_r)P(x_r) \prod_{s \in r^+} P(x_{a(s)}, y_{a(s)}|x_r) \\ &\quad \text{separation property on tree.} \end{aligned} \quad (8)$$

This factorization permits to split the maximization of the joint distribution w.r.t. the entire set x of hidden variables into separate maximizations w.r.t. to x_r , and $x_{d(s)}$, $s \in r^+$

$$\begin{aligned} & \max_x P(x, y) \\ &= \max_{x_r} \left\{ P(y_r|x_r)P(x_r) \prod_{s \in r^+} \underbrace{\max_{x_{d(s)}} P(x_{d(s)}, y_{d(s)}|x_r)}_{\triangleq \mathcal{P}_s(x_r)} \right\}. \end{aligned}$$

A similar factorization can be used recursively to separate variable x_s from other variables in the maximization w.r.t. $x_{d(s)}$. This yields the following bottom-up recursion of maximizations

$$\begin{aligned} \forall s \neq r, \mathcal{P}_s(x_{s-}) &\triangleq \max_{x_{d(s)}} P(x_{d(s)}, y_{d(s)}|x_{s-}) \\ &= \max_{x_s} \left\{ P(y_s|x_s)P(x_s|x_{s-}) \right. \\ &\quad \left. \cdot \prod_{t \in s^+} \underbrace{\max_{x_{d(t)}} P(x_{d(t)}, y_{d(t)}|x_s)}_{\mathcal{P}_t(x_s)} \right\} \end{aligned}$$

with the maximizer in x_s being a function of parent variable x_{s-}

$$x_s^*(x_{s-}) \triangleq \arg \max_{x_s \in \Lambda} \left\{ P(y_s|x_s)P(x_s|x_{s-}) \prod_{t \in s^+} \mathcal{P}_t(x_s) \right\}. \quad (9)$$

If MAP component \hat{x}_{s-} is known for the parent of s , then this function allows one to deduce the MAP estimate for s , which is $x_s^*(\hat{x}_{s-})$. Hence, as in the standard Viterbi algorithm, the whole MAP estimate \hat{x} can be recovered component by component, in a top-down pass where one has simply to read look-up tables which have been built during the bottom-up sweep according to (9). The Viterbi algorithm on the quadtree is thus conducted in two passes that are summarized in Fig. 3.

The simplified notations $\mathcal{P}_s(x_{s-})$ and $x_s^*(x_{s-})$ should not conceal that both functions also depend on data vector $y_{d(s)}$, supported by s and its descendants.

In practice, the quantities \mathcal{P}_s may be so small that the usual precision of computers is not sufficient (*underflow* is a common problem in Viterbi algorithms). The whole procedure is thus implemented by computing the logarithm of the probabilities (sums of possibly large negative numbers are thus handled instead of products of tiny positive factors).

B. MPM Estimation

It is well known that MAP cost function, which penalizes the discrepancies between configurations without any consideration about how much different these configurations are, provides an

Two-pass MAP estimation on the quad-tree

▲ **Bottom-up pass.** The upward recursive sweep consists in computing and storing functions $\mathcal{P}_s(x_{s-})$ and $x_s^*(x_{s-})$ for all possible values of argument x_{s-} :

$$\begin{aligned} & \text{initialization (leaves } s \in S^N) \\ & \left| \begin{aligned} \mathcal{P}_s(x_{s-}) &= \max_{x_s} P(y_s|x_s)P(x_s|x_{s-}) \\ x_s^*(x_{s-}) &= \arg \max_{x_s} P(y_s|x_s)P(x_s|x_{s-}) \end{aligned} \right. \end{aligned}$$

$$\begin{aligned} & \text{recursion (} s \in S^{N-1} \dots S^1) \\ & \left| \begin{aligned} \mathcal{P}_s(x_{s-}) &= \max_{x_s} P(y_s|x_s)P(x_s|x_{s-}) \prod_{t \in s^+} \mathcal{P}_t(x_s) \\ x_s^*(x_{s-}) &= \arg \max_{x_s} P(y_s|x_s)P(x_s|x_{s-}) \prod_{t \in s^+} \mathcal{P}_t(x_s) \end{aligned} \right. \end{aligned}$$

▼ **Top-down pass.** The downward sweep consists in successively recovering the components of the MAP estimate, starting from the root, down to the leaves:

$$\begin{aligned} & \text{initialization (root } r) \\ & \hat{x}_r = \arg \max_{x_r} P(y_r|x_r)P(x_r) \prod_{s \in r^+} \mathcal{P}_s(x_r) \\ & \text{recursion (} s \in S^1 \dots S^N) \\ & \hat{x}_s = x_s^*(\hat{x}_{s-}) \end{aligned}$$

Fig. 3. Two-pass MAP estimation on the quadtree.

estimator that may exhibit undesirable properties. The following cost function is generally better behaved:

$$\forall (x, x') \in \Omega \times \Omega, C(x, x') = \sum_{s \in S} \delta(x_s, x'_s). \quad (10)$$

The resulting Bayesian estimator is the mode of posterior marginals (MPM) estimator which associates to each site the most probable value given all the data

$$\forall s \in S, \hat{x}_s = \arg \max_{x_s \in \Lambda} P(x_s|y). \quad (11)$$

This estimator requires the computation of the posterior marginals $P(x_s|y)$ from the original joint distribution $P(x, y)$. This is generally a difficult issue since each of these functions should be obtained by simultaneously integrating out all x_t , $t \neq s$. However, the tree structure allows once again to design a noniterative method to solve the problem.

The standard two-sweep “forward-backward” algorithm which has been introduced by Baum *et al.* for chain-based models [2], can be directly extended to trees. Different versions of such an extension have been introduced in the context of so-called graphical models and belief networks (used in multivariate statistics and artificial intelligence) [26], [27], [33], [39], [45], [48], as well as in signal processing domain [9].

Unfortunately, we found them difficult to use for the large image inverse problems we are dealing with. As a fact, the first propagation sweep that they are all based on, recursively computes subtree data likelihoods of the form $P(y_{d(s)}|x_s)$. In case of large quadtrees, the number of data components in $y_{d(s)}$ rapidly grows as the upward sweep proceeds, yielding probabilities that are so small, that their practical manipulation on computer becomes difficult (underflow problem). In some of the above mentioned algorithms, this problem risks also to plague the downward recursion (e.g., when it is based on joint laws of type $P(x_s, y_{S \setminus d(s)})$ as in [9]). Although it is possible to design proper

normalization to alleviate the difficulty when using such procedures [41], we now introduce an alternative two-sweep procedure that allows to compute posterior marginals in a “safe” way.

The starting point of our original procedure lies in the expression of the posterior marginal $P(x_s|y)$ as a function of the posterior marginal at parent node x_{s-}

$$\begin{aligned} P(x_s|y) &= \sum_{x_{s-}} P(x_s|x_{s-}, y)P(x_{s-}|y) \\ &= \sum_{x_{s-}} P(x_s|x_{s-}, y_{d(s)})P(x_{s-}|y) \\ &= \sum_{x_{s-}} \frac{P(x_s, x_{s-}|y_{d(s)})}{\sum_{x_s} P(x_s, x_{s-}|y_{d(s)})} P(x_{s-}|y). \end{aligned} \quad (12)$$

This yields a top-down recursion provided that the posterior marginal at the root node $P(x_r|y)$, as well as probabilities $P(x_s, x_{s-}|y_{d(s)})$ are made available. This is achieved by a preliminary upward sweep based on

$$P(x_s, x_{s-}|y_{d(s)}) = P(x_{s-}|x_s)P(x_s|y_{d(s)}). \quad (13)$$

The first factor (corresponding to the prior child-parent probability transition) on the right hand side is easily derived from $P(x_{s-}|x_s) = (P(x_s|x_{s-})P(x_{s-}))/P(x_s)$, where $P(x_s|x_{s-})$ is part of the prior specification, and the prior marginals $P(x_s)$ are computed using a simple top-down recursion: $P(x_s) = \sum_{x_{s-}} P(x_s|x_{s-})P(x_{s-})$.

An upward recursion allows to compute the partial posterior marginals $P(x_s|y_{d(s)})$ in (13)²

$$\begin{aligned} P(x_s|y_{d(s)}) &\propto P(x_s, y_{d(s)}) \\ &= \sum_{x_{s+}} P(y_{d(s)}|x_s, x_{s+})P(x_{s+}|x_s)P(x_s) \\ &= \sum_{x_{s+}} P(y_s|x_s)P(x_s) \prod_{t \in s^+} [P(y_{d(t)}|x_t)P(x_t|x_s)] \\ &\propto P(y_s|x_s)P(x_s) \prod_{t \in s^+} \sum_{x_t} \left[\frac{P(x_t|y_{d(t)})}{P(x_t)} P(x_t|x_s) \right] \end{aligned} \quad (14)$$

where “ \propto ” means that equality holds up to a multiplicative quantity which does not depend on x_s . Note that the product over the children set is actually absent at the leaves of the tree ($s \in S^N$), i.e., at the recursion start. The final result is obtained up to a normalization constant which is easily computed since one deals with single-variable distributions over a finite state space. At the root, the complete posterior marginal $P(x_r|y)$ is eventually obtained, and, on the way up to the root all site-wise and pair-wise partial posterior marginals $P(x_s|y_{d(s)})$ and $P(x_{s-}, x_s|y_{d(s)})$ are computed using (13) and (14). The whole procedure is summarized in Fig. 4.

C. Sequential MAP

Although the MPM criterion seems to be more appropriate than the MAP criterion in terms of underlying cost functions,

Two-pass computation of posterior marginals and MPM estimation on the tree

■ **Preliminary pass.** This downward recursion computes the prior marginal at each node (it is sometimes straightforward, see Section IV.D)

$$s \in S^{N-1} \dots S^1, P(x_s) = \sum_{x_{s-}} P(x_s|x_{s-})P(x_{s-})$$

▲ **Bottom-up pass.** This upward recursion, akin to Kalman filtering, provides the distribution of each x_s (and couple (x_s, x_{s-})) given all the data of the descendants (including y_s):

$$\begin{aligned} &\text{initialization (leaves } s \in S^N) \\ &\left| \begin{aligned} P(x_s|y_s) &\propto P(y_s|x_s)P(x_s) \\ P(x_s, x_{s-}|y_s) &= \frac{P(x_s|x_{s-})P(x_{s-})P(x_s|y_s)}{P(x_s)} \end{aligned} \right. \end{aligned}$$

$$\begin{aligned} &\text{recursion (} s \in S^{N-1} \dots S^0) \\ &\left| \begin{aligned} P(x_s|y_{d(s)}) &\propto P(y_s|x_s)P(x_s) \prod_{t \in s^+} \sum_{x_t} \frac{P(x_t|y_{d(t)})P(x_t|x_s)}{P(x_t)} \\ P(x_s, x_{s-}|y_{d(s)}) &= \frac{P(x_s|x_{s-})P(x_{s-})P(x_s|y_{d(s)})}{P(x_s)} \quad \forall s \neq r \end{aligned} \right. \end{aligned}$$

▼ **Top-down pass.** This downward recursion re-assembles the complete posterior marginals from the partial marginals computed in the bottom-up sweep, and deduces MPM components:

$$\begin{aligned} &\text{initialization} \\ &\left| \begin{aligned} P(x_r|y) &= P(x_r|y_{d(r)}) \\ \hat{x}_r &= \arg \min_{x_r} P(x_r|y) \end{aligned} \right. \end{aligned}$$

$$\begin{aligned} &\text{recursion} \\ &\left| \begin{aligned} P(x_s|y) &= \sum_{x_{s-}} \frac{P(x_s, x_{s-}|y_{d(s)})}{\sum_{x_{s-}} P(x_s, x_{s-}|y_{d(s)})} P(x_{s-}|y) \\ \hat{x}_s &= \arg \min_{x_s} P(x_s|y) \end{aligned} \right. \end{aligned}$$

Fig. 4. Two-pass computation of posterior marginals and MPM estimation on the tree.

MPM and MAP cost functions do not take into account the location of estimation errors in the hierarchical quadtree structure. Bouman and Shapiro introduced the following cost function [4]:

$$\begin{aligned} &\forall (x, x') \in \Omega \times \Omega, \\ &C(x, x') = \sum_{n=0}^N 2^{N-n} \left(1 - \prod_{k=0}^n \delta(x^k, x'^k) \right) \end{aligned} \quad (15)$$

where term $1 - \prod_{k=0}^n \delta(x^k, x'^k)$ is exactly the MAP cost function applied only to levels 0 to n . The estimator associated to this weighted combination of partial MAP cost functions has been named “sequential MAP.” The higher a node on the tree, the more numerous penalty terms it is involved in. Penalties thus increase when the resolution decreases, which seems to be a sensible requirement.

Contrary to the standard MAP estimator, the novel estimator defined by cost function (15) is however not easy to explicit. Bouman and Shapiro propose a noniterative inference

²Since this upward recursion propagates partial posterior marginals $P(x_s|y_{d(s)})$, which are univariate distributions, the underflow problem evoked at the beginning of Section IV-B is thus not encountered here. Note that our two-sweep algorithm can be seen as the exact discrete analog of two-sweep smoothing RTS algorithm for Gaussian state-space models (introduced by Rauch *et al.* for chain-based dynamical models [43], and then extended by Chou *et al.* on tree-based dynamical models [7]), where Kalman-type filtering propagates normal distributions $P(x_s|y_{d(s)})$ within the upward pass.

algorithm for computing approximate SMAP estimates, in the case of scalar data defined at the finest resolution only. This algorithm is easily extended in our case where data vectors may be available at all resolutions. The inference is then performed with the following top-down recursion:

$$\begin{aligned}\hat{x}_r &\approx \arg \max_{x_r} P(x_r|y) \\ \hat{x}_s &\approx \arg \max_{x_s} P(x_s|\hat{x}_{s-}, y_{d(s)}) \\ &= \arg \max_{x_s} P(y_{d(s)}|x_s)P(x_s|\hat{x}_{s-})\end{aligned}$$

where the root posterior marginal can be obtained as explained in Section IV-B. The multivariate conditional likelihoods $P(y_{d(s)}|x_s)$ can also be computed in a preliminary bottom-up sweep, since

$$P(y_{d(s)}|x_s) = P(y_s|x_s) \prod_{t \in t^+} \sum_{x_t} P(y_{d(t)}|x_t). \quad (16)$$

This is the extension to trees of “forward” sweep designed by Baum *et al.* on chains [2]. As discussed in Section IV-B, these computations are unfortunately plagued by underflow problems since the probabilities become extraordinary small as the number of data components increases. We suggest to use instead the upward recursion that we have introduced for the MPM estimation (see Section IV-B). This recursion actually provides the distributions $P(x_s, x_{s-}|y_{d(s)})$ from which $P(x_s|x_{s-}, y_{d(s)})$ is easily recovered by normalization.

D. Experimental Results: Supervised Classification

In order to validate the different estimation algorithms on the quadtree and to get some insight into their properties, we have conducted a number of experiments in image classification. Supervised classification aims at assigning the observed pixels to predefined classes, based on intensity or texture criteria. A class is associated to a region of the image plane which is not necessarily connected, but in which intensities share a similar statistical behavior (in terms of some prescribed model). We have chosen a simple model, where each class is characterized by a Gaussian model, defined by a mean vector and a variance-covariance matrix. For a same class, these parameters can be different from one resolution to another. Each class $i \in \Lambda = \{1 \dots M\}$ is then defined by $(\mu_i^n, \Gamma_i^n)_{n=0}^N$ where $\mu_i^n \in \mathbb{R}^{d_n}$ is the mean vector at level n (data is d_n -dimensional at that level), and Γ_i^n designates the associated variance-covariance matrix. The point-wise conditional likelihoods are represented by

$$\begin{aligned}\forall s \in S^n, P(Y_s = y_s|X_s = i) \\ = \frac{1}{\sqrt{\det(2\pi\Gamma_i^n)}} \exp \left\{ -\frac{1}{2}(y_s - \mu_i^n)^T (\Gamma_i^n)^{-1} (y_s - \mu_i^n) \right\}\end{aligned}$$

where the data are assumed to lie in $\Upsilon = \mathbb{R}$, although they are in practice within the discrete set $\{0, \dots, 255\}$. We assume here that the number of classes M is known, and that the parameters associated to each class are obtained by some preliminary supervised learning step.

For the prior distribution on the quadtree, we have adopted the Potts-like distribution used by Bouman and Shapiro [4]. This

TABLE I
SUPERVISED CLASSIFICATION RESULTS ON
IMAGES FIGS. 5(a) AND 6(a)

Algorithm	Success		Cpu time	
	Fig. 5a	Fig. 6a	Fig. 5a	Fig. 6a
NH-ICM	89%	57%	10 s	11 s
NH-MAP	97.1%	87%	444 s	708 s
H-MAP	96%	91%	26 s	28 s
H-SMAP	97.4%	89%	36 s	19 s
H-MPM	95.5%	93%	16 s	9 s

simple model favors identity between parent and children, all other transitions being equally (un)likely

$$\forall s \in S^n, P(X_s = j|X_{s-} = i) \triangleq \begin{cases} \theta_n, & \text{if } i = j, \\ \frac{1 - \theta_n}{M - 1}, & \text{otherwise} \end{cases} \quad (17)$$

with $\theta_n > 1/M$, and uniform prior is chosen at root r . For easier parameter tuning, we actually kept θ_n independent from level n in our experiments. The value of the unique prior parameter θ was then set to 0.9. Note that for this in-scale homogeneous prior with a single parameter, marginals are obviously uniform at all nodes of the tree. Hence, the preliminary sweep from the posterior marginal computation algorithm becomes unnecessary, and computations in both upward and downward sweeps are simplified by equating all $P(x_s)$ with $1/M$.

In the following, we denote H-MAP (hierarchical MAP) the exact MAP estimate associated to the model on the quadtree and obtained as explained in Section IV-A. Similarly, H-MPM stands for the exact MPM estimate, computed as shown in Section IV-B, and H-SMAP stands for the generalization of the SMAP presented in Section IV-C. In the case of single-resolution data, we can compare these three estimates with the approximated MAP estimate of a standard lattice-based classification model. This lattice-based model is defined with the same data likelihood as the hierarchical representation, and is based on a Potts prior on a first-order neighborhood [19]. This nonhierarchical MAP estimate can be obtained iteratively either by simulated annealing (we denote the resulting estimate by NH-MAP), or by a deterministic ICM algorithm [3] whose final classification will be referred to as NH-ICM. These two nonhierarchical iterative algorithms are stopped when the number of actual updates, after a complete sweep of the image, falls below a given threshold (one per 1000 of the total number of pixels). The cooling schedule in the simulated annealing procedure is defined as $T_k = T_0/(1.01)^k$ where T_0 is the initial temperature, set to 100 and k stands for the current number of image sweeps.

The performances of the different methods are first evaluated on synthetic images with known parameters (i.e., the number of classes and the parameters of each class) and ground-truth. In this case, only one full resolution scalar data image is used. We report the rates of correct classification and the required cpu times on a SunSparc 10 workstation (see Table I). A first 256×256 synthetic image [Fig. 5(a)] is composed of disks with various radii in front of a homogeneous background. There are five classes with different means (50, 76, 105, 149, and 178) and the same variance 937. This corresponds to a SNR of -0.27 dB

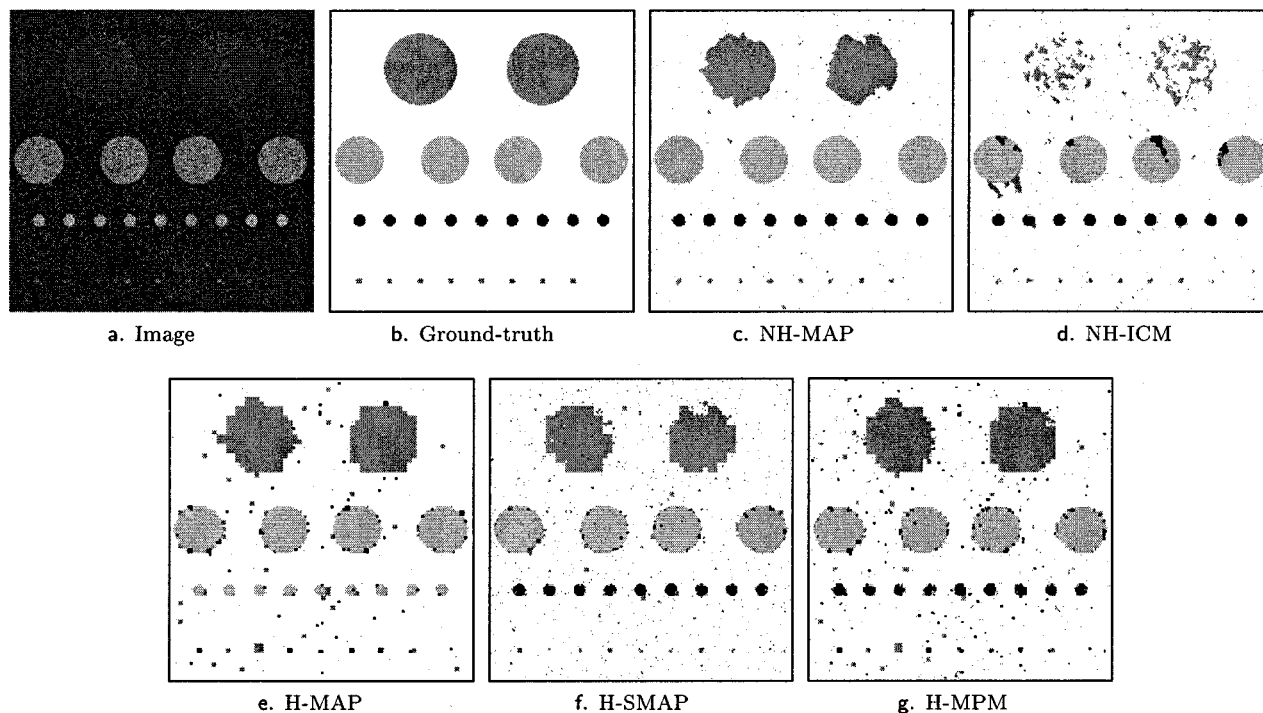


Fig. 5. (a) Image 256×256 (SNR -0.27 dB; for the sake of readability, printed gray levels do not correspond to the actual means); (b) ground truth classification; and classifications obtained by (c) NH-MAP, (d) NH-ICM, (e) H-MAP, (f) H-SMAP, and (g) H-MPM.

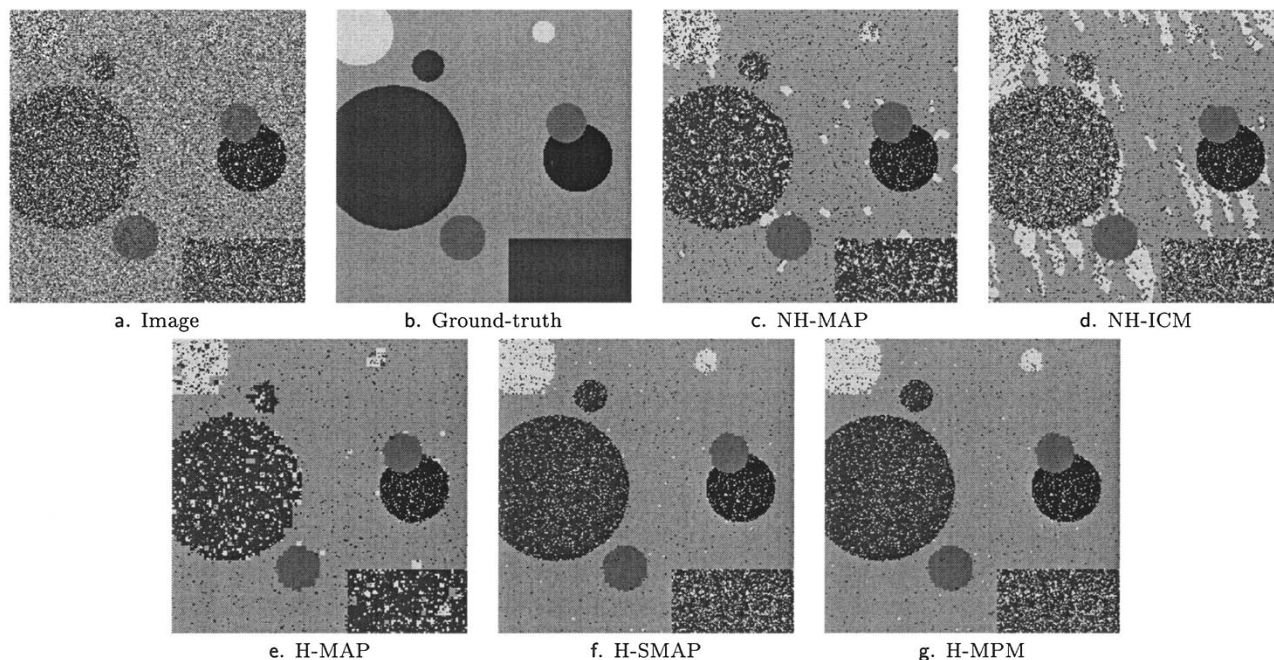


Fig. 6. (a) Image 256×256 (SNR -2.16 dB; for the sake of readability, gray levels do not correspond to the actual means); (b) ground truth classification; and classifications obtained by (c) NH-MAP, (d) NH-ICM, (e) H-MAP, (f) H-SMAP, and (g) H-MPM.

if the image is considered as composed of constant gray level regions corrupted by additive noise. A second 256×256 image [Fig. 6(a)] also consists of five classes ($M = 5$) with means 20, 50, 100, 150, 210, and variances 67, 74, 20, 50, and 77, respectively (the SNR is -2.16 dB).

As expected, the deterministic nonhierarchical ICM method is very sensitive to noise, and shows fast convergence toward poor estimates. For both images, this method provides the

worse rate of correct classification. Its stochastic counterpart, NH-MAP, behaves quite well for low noise levels, but tends to “over-smooth” the estimate as the level of noise increases (as often noticed). In any case, it is, by far, the slowest inference procedure. Better results could probably be obtained with NH-MAP by using slower temperature schedules, but this would result in an even longer estimation time. The three noniterative estimators on the quadtree provide a good compromise

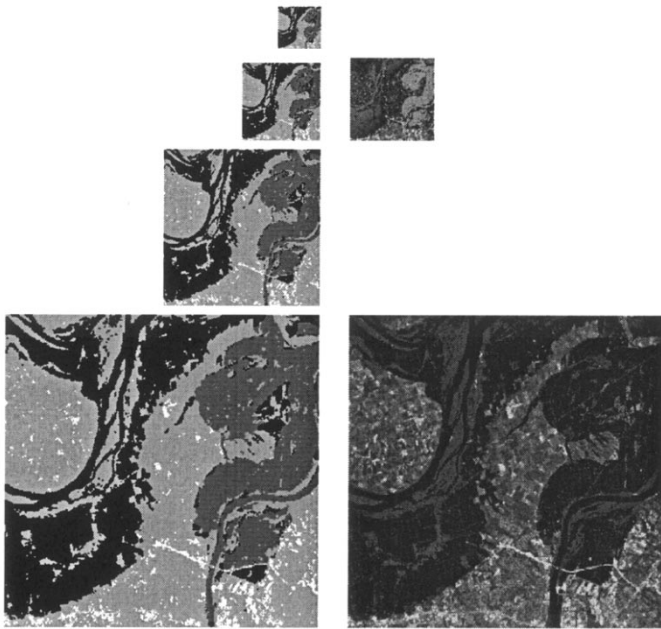


Fig. 7. Hierarchical classification of multiresolution airborne images with the Viterbi algorithm on the quadtree structure (H-MAP). The two data images are on the right, and the classification obtained on the last four levels of the quadtree are on the left. CPU time: 1 min 40 s.

between the quality of the results and the computational load. In particular, on the second image, they provide the best classifications compared to the two nonhierarchical inferences, and are 30 or 80 times faster than simulated annealing.³ Not surprisingly, block artifacts can be noticed in the classifications provided by quadtree-based inferences. Although visually disturbing, these artifacts have no real impact on the correct classification rate. The boundaries of regions are also poorly recovered by nonhierarchical methods, but the less structured nature of errors makes them less noticeable at the first glance.

To illustrate the ability of hierarchical models to deal with multiresolution data, we have also implemented another classification experiment on real airborne images. The scene, obtained in the visible range, represents the Mississippi and the Missouri rivers during the historical flood of June 1993. A classification into four classes ($M = 4$) has been considered: one class for each river, one class for the urban areas, and a fourth class for forests and swamps. Apart from its semantic meaning, this classification is supported by homogeneity properties observed within each class. Two resolutions levels were created, corresponding to images of sizes 512×512 and 128×128 (right side of Fig. 7). The three hierarchical algorithms inferred the classifications from the two data sets at levels S^9 and S^7 of the quadtree and provided comparable results. We only display in Fig. 7 the classes obtained with the Viterbi algorithm (corresponding to an exact MAP estimate on the quadtree). As can be

³The noniterative nature of algorithms on quadtree amounts to a constant per-point complexity whereas for iterative inference with grid-based models the per-point complexity grows with grid size. More precisely, the total complexity of one of the two-sweep algorithms on a m -leave quadtree is $\mathcal{O}(M(4m - 1/3))$, whereas the complexity of typical iterative inference on m -site grid S^0 is $\mathcal{O}(mM)$ per iteration, the average number of required iterations being an increasing function of the sizes of state space Λ and of finest resolution image support S^0 .

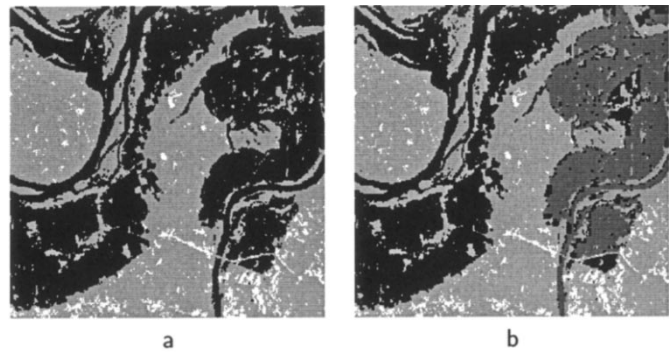


Fig. 8. (a) H-MAP based on the finest resolution image only and (b) H-MAP based on multiresolution data. The two rivers are now discriminated.

seen, the blocky artifacts are hardly visible on these real-world images. The impact of data fusion is demonstrated by comparing these results with the four-class classifications obtained with only one image [Fig. 8(a)]. The fact that the two rivers are quite distinct on a low-resolution image allowed to keep them distinctly classified *at all the levels* of the tree, including the finest ones, although they are hardly distinguishable on finest resolution image. Using only the latter image does not allow to discriminate the two rivers. This simple experiment indicates; i) that the most discriminant information (if any) can be accessed at whatever resolution it is and ii) that this information can then be used within tree-based framework to improve statistical inference all over the structure (not only at or around concerned resolution). As for computation load, the H-MAP took 1 min 40 s on the same workstation as before.

V. ESTIMATING PARAMETERS ON THE QUADTREE

In the previous section, we have assumed that both the prior parameters (root prior distribution, parent-child transition probabilities) and the data likelihood parameters (variances and expectations for each Gaussian class) were known for the model on the quadtree. We have thus dealt with *supervised* inference problems. However, since the exact value of these parameters is often critical, and since their manual tuning is usually difficult, automatic methods for estimating the model parameters are highly desirable.

It is known that estimating parameters of probabilistic models in large inverse problems is generally involved, because only Y is observed, while X remains hidden (this is often called the “incomplete data” problem). To cope with this problem, the standard maximum likelihood framework has been extended to the incomplete data case, through the expectation-maximization (EM) algorithm [44]. The EM algorithm is an iterative procedure that increases the likelihood of observed data at each step, based on the posterior distribution relative to the previous parameter estimate. These techniques are quite convenient in the case of mixtures of distributions for which they have been originally designed. However, they turn out to be extremely cumbersome for standard lattice-based MRF models [6]. As we shall explain, the hierarchical model on the quadtree allows to dramatically alleviate this problem, without reducing the quality of the estimates. Note that Bouman and Shapiro have proposed an EM algorithm on the quadtree model, for the estimation of the

single parameter of their specific prior classification model [4]. In the following, we introduce a comprehensive procedure for estimating all prior and data likelihood parameters of any discrete model on the quadtree.

A. Background on EM-Type Algorithms

Let Φ be the parameter vector involved in the joint distribution denoted $P(x, y|\Phi)$. The likelihood of data is defined as $\mathcal{L}(\Phi) \triangleq \log P(y|\Phi)$, and the maximum likelihood estimate (MLE) of Φ is then

$$\hat{\Phi} \triangleq \arg \max_{\Phi} P(y|\Phi). \quad (18)$$

The data likelihood is in general not computable [due to a summation over all possible configurations: $P(y) = \sum_x P(x, y)$], or at least not available in closed form as a function of Φ . For that reason, EM uses the expectation of the joint likelihood function $\log P(x, y|\Phi)$ taken w.r.t. the posterior distribution, with the current parameter fit $\Phi^{(k)}$

$$Q(\Phi|\Phi^{(k)}) \triangleq \mathbb{E} \left[\log P(X, y|\Phi) | y, \Phi^{(k)} \right] \\ = \sum_{x \in \Omega} P(x|y, \Phi^{(k)}) \log P(x, y|\Phi). \quad (19)$$

The new estimate is then ideally chosen by maximizing function Q , yielding the genuine EM step

- Expectation (E) step: compute function $Q(\Phi|\Phi^{(k)})$;
- Maximization (M) step: update

$$\Phi^{(k+1)} = \arg \max_{\Phi} Q(\Phi|\Phi^{(k)}). \quad (20)$$

It can be shown that this iterative procedure ensures that the data likelihoods $\mathcal{L}(\Phi^{(k)})$ are increasing: the procedure converges toward a local maxima or a saddle point of the likelihood function [44]. Unfortunately, the estimate at convergence is generally highly dependent on the initial parameter fit and convergence is usually very slow. Apart from these problems, two other technical difficulties arise, which make the implementation of the EM steps difficult: 1) the joint distribution $P(x, y|\Phi)$ is usually only known up to a normalization constant that depends on Φ and 2) the computation of the conditional expectation by summing over all possible configurations x is generally intractable.

Attempts to cope with the above mentioned problems have originated several variants of the original EM algorithm. In order to avoid dependence w.r.t. the initial estimate, a stochastic version of the EM algorithm called SEM has been proposed (see [5] for a review). Besides, the computation of the expectation may be done approximately using Monte Carlo Markov Chain (MCMC) techniques based on samples drawn from the posterior distribution $P(x|y, \Phi^{(k)})$ [47]. The resulting Monte Carlo EM (MCEM) actually includes SEM as a special case. These methods can become extremely time consuming since

they resort to iterative Monte Carlo procedures within each EM step.

For standard noncausal MRF models, the prior distribution $P(x|\Phi)$ is usually known up to a multiplicative constant which depends on Φ and is not computable. A standard solution to circumvent this problem in the computation of function Q is to replace the joint likelihood $\log P(x, y|\Phi) = \log[P(y|x, \Phi)P(x|\Phi)]$ by the so-called *pseudo-likelihood* $\log[P(y|x, \Phi)P(x|\Phi)]$ with $P(x) \triangleq \prod_s P(x_s|x_{n(s)})$, where $n(s)$ designates the neighborhood of s in the chosen independence graph [6], [49]. A “global” likelihood is thus replaced by a sum of “local” likelihoods. Based on this principle, Chalmond developed a Monte Carlo “Gibbsian-EM” algorithm for image classification [6]. The “Gibbsian-EM” algorithm iteratively estimates the parameters of data likelihoods along with those of the spatial prior model. The Monte Carlo sampling used to approximate the expectations actually yields approximation of both pair-wise and site-wise posterior marginals. The latter approximations can be used to get approximate MPM estimates of x , simultaneously with the estimation of parameters. Due to the slow convergence of the iterative Monte Carlo procedure, the whole procedure is expensive. Besides, the substitution of the likelihood by the pseudo-likelihood does not guarantee the convergence of the procedure, even to a local minima.

The different problems that arise when EM algorithms are applied to standard lattice-based MRF models, are not encountered in the special case of tree-based models where there is no unknown normalization constant (partition function), and local posterior marginals can be computed exactly. For the particular case of models on chains, Baum *et al.* have thus been able to develop a parameter estimation algorithm which makes use of noniterative EM steps [2]. This classical algorithm is now widely used in speech recognition for instance [14], [38], as well as in chain-based image analysis [21]. Based on the extension to trees of Baum’s forward-backward algorithm for computing posterior marginals (see Section IV-B), EM technique has been naturally used for discrete tree-based models in artificial intelligence, multivariate statistics, and signal processing [9], [23], [45]. The EM approach we propose differs from those methods in that it relies at each E -step on the original posterior marginal computation technique introduced in Section V-A. In addition, taking advantage of the sampling facilities offered by tree-based models, we introduce an efficient MCEM algorithm on trees which improves learning performances at a reasonable extra computation cost.

B. EM Algorithm on the Quadtree

Let us consider again the joint model (X, Y) introduced in Section III. Each discrete random variable X_s takes its value in $\Lambda = \{1, \dots, M\}$, while the corresponding measurement Y_s is discrete or continuous, with state-space Υ . Making the dependency w.r.t. to the model parameters explicit, the joint distribution may be expressed as

$$P(x, y|\Phi) = P(x_r|\Phi) \prod_{s \in S \setminus \{r\}} P(x_s|x_{s-}, \Phi) \prod_{s \in S} P(y_s|x_s, \Phi).$$

There is no unknown normalization constant in this case, and the expectation in the E -step reduces to

$$\begin{aligned}
Q(\Phi|\Phi^{(k)}) &= \sum_{i \in \Lambda} P(X_r = i|y, \Phi^{(k)}) \log P(X_r = i|\Phi) \\
&+ \sum_{s \in S \setminus \{r\}} \sum_{i \in \Lambda} \sum_{j \in \Lambda} P(X_s = j, X_{s-} = i|y, \Phi^{(k)}) \\
&\cdot \log P(X_s = j|X_{s-} = i, \Phi) \\
&+ \sum_{s \in S} \sum_{i \in \Lambda} P(X_s = i|y, \Phi^{(k)}) \log P(y_s|X_s = i, \Phi).
\end{aligned} \tag{21}$$

In the case of discrete data state space Υ , the most general parameterization of the model is defined by the different probabilities appearing in the latter factorization:

- prior root probabilities: $\pi_i \triangleq P(X_r = i)$;
- prior parent-child transition probabilities which are assumed to be independent of the resolution level:⁴ $a_{ij} \triangleq P(X_s = j|X_{s-} = i), \forall s \neq r$;
- site-wise conditional data likelihood probabilities, which are here supposed to depend on the resolution level: $f_i^n(\ell) \triangleq P(Y_s = \ell|X_s = i), \forall s \in S^n$.

In this case, the parameter vector to be estimated is $\Phi = ((\pi_i)_{i \in \Lambda}, (a_{ij})_{i, j \in \Lambda}, (f_i^n(\ell))_{n=0 \dots N, i \in \Lambda, \ell \in \Upsilon})$, with the constraints:

$$\sum_{i \in \Lambda} \pi_i = 1; \forall i \in \Lambda, \sum_{j \in \Lambda} a_{ij} = 1; \forall i \in \Lambda, \sum_{\ell \in \Upsilon} f_i^n(\ell) = 1. \tag{22}$$

The expectation in the E -step (21) becomes:

$$\begin{aligned}
Q(\Phi|\Phi^{(k)}) &= \sum_{i \in \Lambda} \zeta_r^{(k)}(i) \log \pi_i \\
&+ \sum_{s \in S \setminus \{r\}} \sum_{(i, j) \in \Lambda^2} \xi_s^{(k)}(i, j) \log a_{ij} \\
&+ \sum_{n=0}^N \sum_{s \in S^n} \sum_{i \in \Lambda} \zeta_s^{(k)}(i) \log f_i^n(y_s)
\end{aligned} \tag{23}$$

where $\zeta_s^{(k)}(i) \triangleq P(X_s = i|Y = y, \Phi^{(k)})$, and $\xi_s^{(k)}(i, j) \triangleq P(X_s = j, X_{s-} = i|Y = y, \Phi^{(k)})$.

With this discrete setting, it is possible to implement the *exact* EM algorithm, i.e., both the expectation and the maximization may be conducted without any approximation. This requires the exact computation of site-wise and pair-wise

⁴This stationary assumption for the causal prior could be discarded by making transition probabilities depend on concerned level n , akin to data likelihoods. This however seemed to us as not desirable in practice, due first to the rather reduced amount of information on which each of these parameters would then be based (at least for n close to N , i.e., for smaller levels), and second to the increase of complexity that would result from this over-parameterization. Hence, we preferred to keep the parameterization reasonably parsimonious by not using this degree of freedom.

posterior marginals appearing in (23). We have already seen in Section IV-B how the site-wise posterior marginals $P(x_s|y)$ may be computed within two passes on the tree.⁵ The pair-wise marginals are obtained from the same downward computation (12)

$$P(x_s, x_{s-}|y) = \frac{P(x_s, x_{s-}|y_{d(s)})}{\sum_{x_s} P(x_s, x_{s-}|y_{d(s)})} P(x_{s-}|y).$$

For a current parameter fit $\Phi^{(k)}$, it is thus easy to get $\zeta_s^{(k)}(i)$, and $\xi_s^{(k)}(i, j)$ and then to perform the maximization of $Q(\Phi|\Phi^{(k)})$ subject to the constraints (22), using Lagrange multiplier techniques. One gets the following M -step update:

$$\forall (i, \ell) \in \Lambda \times \Upsilon, \quad f_i^{n(k+1)}(\ell) = \frac{\sum_{s \in S^n: y_s = \ell} \zeta_s^{(k)}(i)}{\sum_{s \in S^n} \zeta_s^{(k)}(i)}, \tag{24}$$

$$\forall (i, j) \in \Lambda \times \Lambda, \quad \begin{cases} a_{ij}^{(k+1)} = \frac{\sum_{s \neq r} \xi_s^{(k)}(i, j)}{\sum_{s \neq r} \zeta_{s-}^{(k)}(i)} \\ \pi_i^{(k+1)} = \zeta_r^{(k)}(i). \end{cases} \tag{25}$$

In unsupervised experiments (Section IV-D), we will use prior parameterization involved in (25). Note, however, that for the simplified prior model used in Section IV-D where prior parent-child probability transitions (17) are defined by a single parameter θ_n at level n , previous E -step is readily adapted. The constrained maximization of Q w.r.t. θ_n provides the following updating in this case:

$$\theta_n^{(k+1)} = \frac{\sum_{s \in S^n} \sum_{i \in \Lambda} \xi_s^{(k)}(i, i)}{|S^n| |\Lambda|} \tag{26}$$

instead of (25). If in addition θ_n is kept independent from level n , then the update of unique parameter θ is

$$\theta^{(k+1)} = \frac{\sum_{s \in S \setminus \{r\}} \sum_{i \in \Lambda} \xi_s^{(k)}(i, i)}{(|S| - 1) |\Lambda|}. \tag{27}$$

In the case of continuous Gaussian data model we use in our experiments, the estimation of function f_i^n is replaced by the estimation of the mean μ_i^n and covariance matrix Γ_i^n , for each value i of Λ . The update of prior parameters remains unchanged,

⁵As mentioned in footnote 2, the original tree-based marginal computation method introduced in Section IV-B can be seen as the discrete counterpart of the tree-based RTS algorithm. As a consequence, the EM technique we now develop can be seen as a discrete analog of EM procedures for chain-based and tree-based dynamical models [14], [28].

and the update of the Gaussian parameters is obtained by multivariate regression

$$\begin{aligned}\mu_i^{n(k+1)} &= \frac{\sum_{s \in S^n} \zeta_s^{(k)}(i) y_s}{\sum_{s \in S^n} \zeta_s^{(k)}(i)} \\ \Gamma_i^{n(k+1)} &= \frac{\sum_{s \in S^n} \zeta_s^{(k)}(i) \left[y_s - \mu_i^{n(k+1)} \right] \left[y_s - \mu_i^{n(k+1)} \right]^T}{\sum_{s \in S^n} \zeta_s^{(k)}(i)}.\end{aligned}\quad (28)$$

The two-pass posterior marginal computations and parameter updates are iterated until convergence is reached. Initialization and convergence criteria will be discussed in Section V-C. Once the parameters are estimated, the inference of x given y can be conducted according to one of the methods described in Section IV, but the MPM estimator is preferred since the posterior marginals are available as a by-product of the EM algorithm.

C. MCEM Algorithm on the Quadtree

A stochastic version of the EM algorithm may be useful in case of bad convergence of the exact deterministic EM algorithm. The principle of MCEM (which admits SEM as a special case) is to draw samples $x^{(k,1)} \dots x^{(k,R)}$ from the posterior distribution $P(x|y, \Phi^{(k)})$ for the current parameter fit, and then to make estimations based on these samples. More precisely, under proper ergodicity assumptions, the posterior marginals may be estimated by

$$\begin{aligned}P\left(X_s = i | Y = y, \Phi^{(k)}\right) &\approx \frac{1}{R} \sum_{m=1}^R \mathbf{1} \left[x_s^{(k,m)} = i \right], \\ P\left(X_s = j, X_{s-} = i | Y, \Phi^{(k)}\right) &\approx \frac{1}{R} \sum_{m=1}^R \mathbf{1} \left[x_s^{(k,m)} = j, x_{s-}^{(k,m)} = i \right].\end{aligned}$$

Denoting now by $\tilde{\zeta}_s^{(k)}(i)$ and $\tilde{\zeta}_s^{(k)}(i, j)$ these ergodic approximations, the expectation to be minimized may be approximated by

$$\begin{aligned}\tilde{Q}\left(\Phi | \Phi^{(k)}\right) &\triangleq \sum_{i \in \Lambda} \tilde{\zeta}_r^{(k)}(i) \log \pi_i \\ &+ \sum_{s \in S \setminus \{r\}} \sum_{(i, j) \in \Lambda^2} \tilde{\zeta}_s^{(k)}(i, j) \log a_{ij} \\ &+ \sum_{n=0}^N \sum_{s \in S^n} \sum_{i \in \Lambda} \tilde{\zeta}_s^{(k)}(i) \log f_i^n(y_s).\end{aligned}$$

With these notations, the constrained minimization leads to the same update (24) and (25) as for the exact EM algorithm.

The sampling issue remains to be addressed. Using the causal structure of the model, a noniterative causal sampling is possible. The sampling algorithm relies on the causal factorization of the *posterior* distribution $P(x|y)$

$$\begin{aligned}P(x|y) &= P(x_r|y) \prod_{s \neq r} P(x_s | x_{s-}, y) \\ &= P(x_r|y) \prod_{s \neq r} P(x_s | x_{s-}, y_{a(s)}).\end{aligned}\quad (29)$$

On the tree structure, $P(x_r|y)$ and $P(x_s | x_{s-}, y_{a(s)})$ can be computed within one upward recursion, as already explained. Once the different factors of the causal factorization (29) have been obtained, a noniterative sampling algorithm is readily defined by drawing from $P(x_r|y)$, and then in a recursive top-down fashion, from distributions $P(x_s | x_{s-}, y_{a(s)})$, where x_{s-} is known from previous samplings.

Although noniterative, the preliminary computations needed to factorize the posterior distribution (29) induces a significant additional cost in the MCEM method. Heuristics may be used to alleviate this extra cost by implementing an approximate sampling from the posterior distribution. The simplest one, which is often used in Markov chain models, consists in replacing $P(x_s | x_{s-}, y_{a(s)})$ by $P(x_s | x_{s-}, y_s) \propto P(y_s | x_s) P(x_s | x_{s-})$ which is a product of known distributions. This gross simplification (which amounts to taking into account data only at s and at its ancestors, when drawing samples at node s) only makes sense when there are actually data all along the paths joining the root to the leaves. This is uncommon in image analysis problems in which only a few resolution levels generally support data. If, e.g., data are only available at S^N , then only samples at the finest level would be data dependent, all the others being driven by the causal prior. A more sensible heuristic, we actually used in our experiments, consists in building a multiresolution data set from the original data set, at locations on the tree where data are missing. Missing data are recovered by low-pass filtering and down-sampling the original data, and the above mentioned approximation is then applied to the full data set. It is thus possible to produce approximate samples from tree-based model at a moderate cost (compared to Monte Carlo methods, such as the Gibbs sampler used to produce samples from noncausal models). These samples may be used within the MCEM algorithm, which is expected to be more robust than the standard EM algorithm on the same hierarchical structure.

D. Experimental Results: Unsupervised Classification

We present experimental results in unsupervised classification, both on synthetic and natural images. As already explained, the unsupervised classification algorithms estimate the number of classes M , the partition of data into classes, the parameters of the different classes (for a Gaussian model here), as well as the parameters of the underlying prior model. We report the results obtained by EM and MCEM on the quadtree, referred to as H-EM and H-MCEM, respectively. We compare these hierarchical schemes to two standard nonhierarchical unsupervised classification methods, also relying on a Gaussian model of luminance. The first one is the Gibbsian EM approach proposed by Chalmond [6], based on Gibbs sampling and on a pseudo-likelihood approximation of spatial Potts prior (with

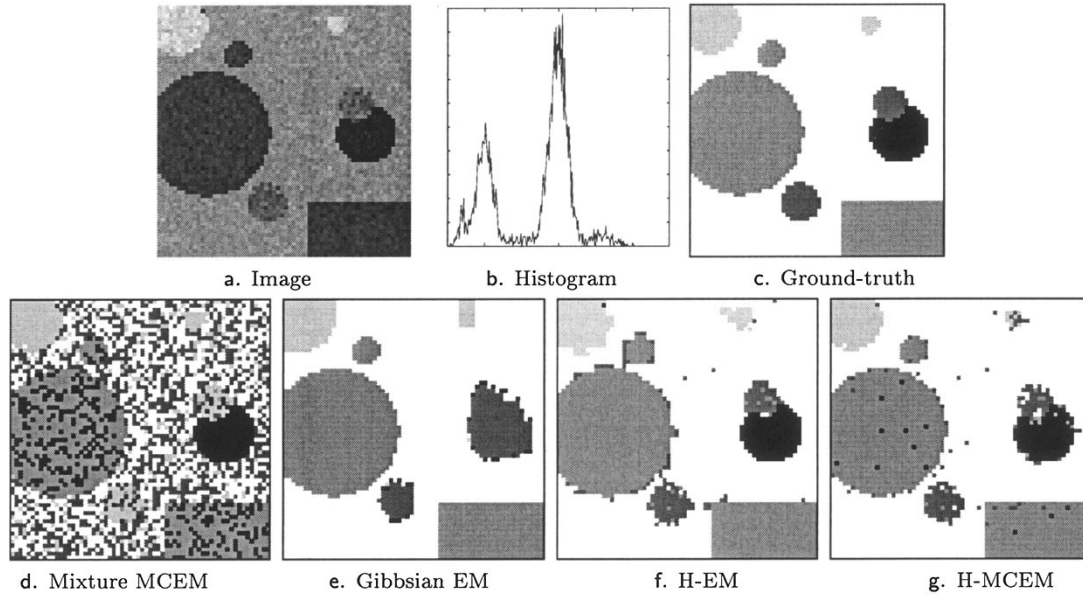


Fig. 9. (a) Synthetic 64×64 image; (b) associated histogram; (c) associated ground-truth (five classes); and (d)–(g) unsupervised classifications obtained with the four EM techniques (the blocky aspect is mainly due to the magnification of these small images).

nearest neighbors interactions); the second one is a plain MCEM method applied to a mixture-based modeling with no interacting prior [47] (i.e., $P(x)$ is taken as $\prod_s P(x_s)$; the prior is then parameterized by the mixture proportions $\{P(X_s = i)\}_{i=1\dots M}$ and the posterior distribution is a product of independent mono-variate distributions). We shall refer to this method as “mixture MCEM.” The comparison with this latter (noncontextual) method will highlight the importance of Markovian prior in the classification task.

The initialization for the class parameters is the same for all algorithms. It is provided by a simple analysis of the finest resolution data histogram. The number of classes, M , is estimated as follows. M is first initialized by a large number; then, at each iteration, classes whose number of occurrences falls under a given threshold are removed, and the number of classes is updated accordingly. For all procedures, the stopping rule is based on the rate of change of the Gaussian likelihood parameters. More precisely, the EM procedures are stopped when $\max_{i,n} (\|\mu_i^{n(k+1)} - \mu_i^{n(k)}\| + \|\Gamma_i^{n(k+1)} - \Gamma_i^{n(k)}\|) < \varepsilon$, with $\varepsilon = 0.1$ in our experiments.

The first test image involves various geometric shapes [Fig. 9(a)]. The luminance within each class follows a scalar Gaussian distribution with means and variances indicated in Table II. The histogram of the image is shown in Fig. 9(b). As can be seen, the histogram only exhibits three or four visible modes, whereas the actual number of classes is five [see the ground-truth in Fig. 9(c)].

The EM algorithms on the quadtree were all able to recover the right number of classes. However, to simplify the comparison between methods, we report the results obtained by the four methods, when run with a number of classes forced to five. The Gaussian parameters estimated for each class are given in Table III. The corresponding MPM classifications are displayed in Fig. 9(d)–(g). Table III shows the computation load of each method, along with the rate of good classification. These results

TABLE II
MEANS AND VARIANCES OF CLASSES ESTIMATED BY THE FOUR EM TECHNIQUES ON IMAGE FIG. 9(a)

Means					
Class	#1	#2	#3	#4	#5
Truth	20	50	100	150	210
mixture	41.4	52.3	141.9	155.2	157.0
Gibbs EM	60.4	49.6	109.4	148.1	197.0
H-EM	21.3	49.5	119.6	149.4	202.1
H-MCM	19.3	49.4	116.7	149.5	210.5
Variances					
Class	#1	#2	#3	#4	#5
Truth	25	100	400	144	225
mixture	280.4	70.3	66.3	54.2	1180.3
Gibbs EM	2255.5	128.8	2150.3	276.8	1441.4
H-EM	182.3	105.5	778.4	138.8	545.2
H-MCM	28.0	102.4	1896.0	134.3	194.9

TABLE III
PERFORMANCES OF THE UNSUPERVISED CLASSIFICATION ALGORITHMS ON IMAGE FIG. 9(a). # ITERATIONS IS THE NUMBER OF EM ITERATIONS TO REACH CONVERGENCE; R IS THE NUMBER OF SAMPLES DRAWN WITHIN EACH SINGLE EM ITERATION

Algorithm	Success	# iter.	R	Cpu time
Mixture MCEM	58%	10	338	26 mn
Gibbsian EM	95%	50	341	3h 12 mn
H-EM	97%	10	none	45 s
H-MCEM	98%	10	228	19 mn

first demonstrate the significant computation saving allowed by algorithms on the quadtree, compared to Gibbsian EM on the lattice. Exact EM on the quadtree is, as expected, the fastest algorithm, since it does not require any sampling. On the other hand, as a deterministic procedure, it requires a good initialization (this was the case here). The two hierarchical methods provide the best results in terms of accuracy of the parameter

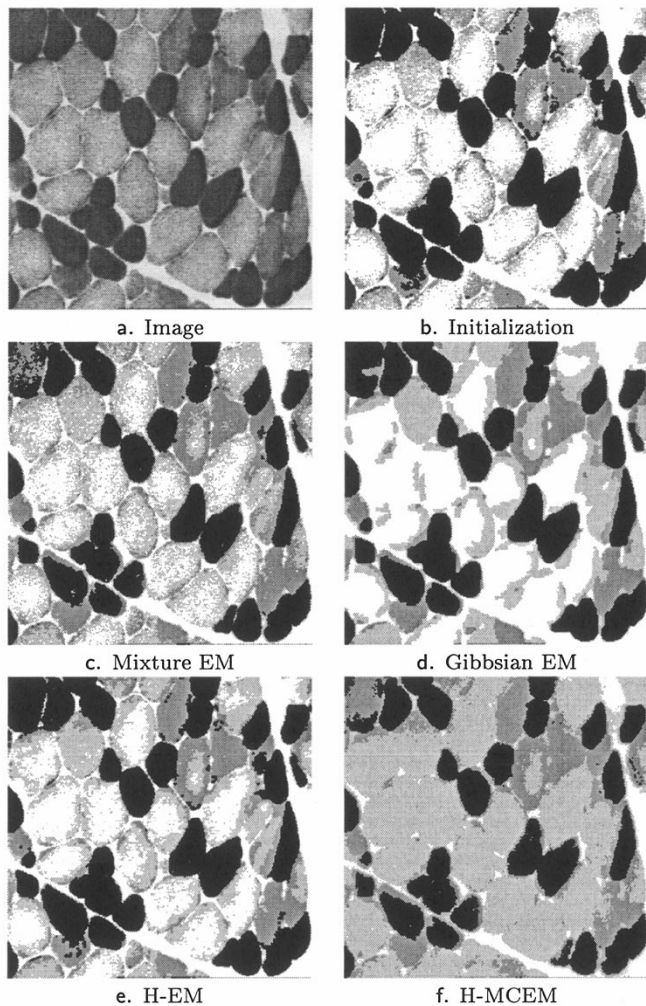


Fig. 10. (a) Original image 256×256 (courtesy of GdR/PRC Isis); (b) maximum likelihood classification used for the initial data parameters estimates; (c) classification with mixture MCEM (# iterations = 10, $R = 52$); (d) classification with Gibbsian EM (# iterations = 50, $R = 41$); (e) classification with EM on the quadtree (# iterations = 3); and (f) classification with MCEM on the quadtree (# iterations = 10, $R = 228$).

estimates and in terms of quality of the associated MPM classification. One can notice, for instance, that Gibbsian EM is not able to discriminate two disks that actually belong to two different classes (see Fig. 9(e) and Table II). As a matter of fact, the estimated variances obtained on classes 1 and 3 with this approach are quite large, while the estimated mean of class 1 is strongly biased. The classifications obtained with the noncontextual mixture EM method are very poor, as expected.

We finally illustrate the hierarchical EM algorithms on a real-world microscope image representing transverse sections of muscle fibers [Fig. 10(a)]. As before, the same initial state has been specified for all algorithms: the number of classes was fixed to four ($M = 4$) and the model parameters were initialized by histogram-based estimation methods. It turns out in this case, that the initialization is poor [see Fig. 10(b)], and both Gibbsian EM and H-EM remain stuck in undesired local minima. These deterministic methods are for instance unable to discriminate the white background from some light cells [Fig. 10(d)–(e)]. This is not the case for the stochastic EM on the quadtree H-MCEM, as

can be seen in Fig. 10(f). This illustrates again the sensitiveness of deterministic EM algorithms to initialization, and shows that the inference may take advantage of the low-cost random sampling provided by H-MCEM. Once again, the noncontextual mixture EM method provides a “noisy” classification, close to the initialization [Fig. 10(c)].

VI. CONCLUSION

We have introduced a family of algorithms for supervised and unsupervised statistical inference on the quadtree. These algorithms, based on nonlinear discrete causal Markov representations, have many potential applications in early vision and inverse imaging problems. Hierarchical MAP, MPM, and SMAP estimators have been developed on the quadtree, as well as EM and MCEM procedures for the unsupervised estimation of the parameters of these models. The performances of these hierarchical inference algorithms have been assessed and compared to standard (noncausal) spatial approaches in an image classification problem. Preliminary experiments have demonstrated that gains may be expected from these new approaches, not only in terms of computation load, but also in terms of estimation quality. The block artifacts, which are induced by the spatial nonstationarity of the quadtree structure, do not seem to be detrimental on real images.

We believe that hierarchical tree-based models could become an appealing alternative to standard Markovian or energy-based models supported by spatial grids. They dramatically reduce the computational load, especially in unsupervised problems, in which noncausal spatial models are often intractable. Besides, these hierarchical models are well suited for the Bayesian processing of multiresolution data. In multiresolution image classification problems, for instance, they enable a consistent fusion of all available data.

Further investigations on these models should deal with more complex hierarchical structures, while preserving the computational advantages of the quadtree. Nonlinear continuous representations (which may arise from the mixing of discrete and Gaussian models on the tree) would also be worth considering in a future work.

ACKNOWLEDGMENT

The authors would like to thank E. Fabre, Irisa/Inria-Rennes, for stimulating discussions.

REFERENCES

- [1] M. Basseville, A. Benveniste, K. C. Chou, S. A. Golden, R. Nikoukhah, and A. S. Willsky, “Modeling and estimation of multiresolution stochastic processes,” *IEEE Trans. Inform. Theory*, vol. 38, pp. 766–784, Mar. 1992.
- [2] L. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains,” *Ann. Math. Stat.*, vol. 41, pp. 164–171, 1970.
- [3] J. Besag, “On the statistical analysis of dirty pictures,” *J. R. Stat. Soc.*, vol. B48, no. 3, pp. 259–302, 1986.
- [4] C. Bouman and M. Shapiro, “A multiscale image model for Bayesian image segmentation,” *IEEE Trans. Image Processing*, vol. 3, pp. 162–177, Feb. 1994.
- [5] G. Celeux, D. Chauveau, and J. Diebolt, “On stochastic versions of the EM algorithm,” INRIA, Tech. Rep. 2514, Mar. 1995.
- [6] B. Chalmond, “An iterative Gibbsian technique for reconstruction of M -ary images,” *Pattern Recognit.*, vol. 22, no. 6, pp. 747–761, 1989.

- [7] K. Chou, S. Golden, and A. Willisky, "Multiresolution stochastic models, data fusion and wavelet transforms," *IEEE Trans. Signal Processing*, vol. 34, pp. 257–282, Mar. 1993.
- [8] M. Comer and E. Delp, "Segmentation of textured images using a multiresolution Gaussian autoregressive model," *IEEE Trans. Image Processing*, vol. 8, pp. 408–420, Mar. 1999.
- [9] M. Crouse, R. Nowak, and R. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 46, pp. 886–902, Apr. 1998.
- [10] K. Daoudi, A. Frakt, and A. Willisky, "Multiscale autoregressive models and wavelets," *IEEE Trans. Inform. Theory*, vol. 45, pp. 828–845, Mar. 1999.
- [11] A. Dawid, "Applications of a general propagation algorithm for probabilistic expert systems," *Stat. Comput.*, vol. 2, pp. 25–36, 1992.
- [12] H. Derin and P. A. Kelly, "Discrete-index Markov-type random processes," *Proc. IEEE*, vol. 77, pp. 1485–1509, Oct. 1989.
- [13] X. Descombes, M. Sigelle, and F. Prêteux, "Estimating Gaussian Markov random field parameters in a nonstationary framework: Application to remote sensing imaging," *IEEE Trans. Image Processing*, vol. 8, pp. 490–503, Apr. 1999.
- [14] V. Digalakis, J. Rohlicek, and M. Ostendorf, "ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition," in *IEEE Trans. Speech Audio Processing*, vol. 1, Sept. 1993, pp. 431–442.
- [15] P. Fieguth, W. Karl, and A. Willisky, "Efficient multiresolution counterparts to variational methods for surface reconstruction," *Comput. Vis. Image Understand.*, vol. 70, no. 2, pp. 157–176, 1998.
- [16] P. Fieguth, W. Karl, A. Willisky, and C. Wunsch, "Multiresolution optimal interpolation and statistical analysis of Topex/Poseidon satellite altimetry," *IEEE Trans. Geosci. Remote Sensing*, vol. 33, pp. 280–292, Feb. 1995.
- [17] P. Fieguth and A. Willisky, "Fractal estimation using models on multiscale trees," *IEEE Trans. Signal Processing*, vol. 44, pp. 1297–1299, May 1996.
- [18] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.
- [19] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 721–741, June 1984.
- [20] B. Gidas, "A renormalization group approach to image processing problems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 2, pp. 164–180, Nov. 1989.
- [21] N. Giordana and W. Pieczynski, "Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 465–475, May 1997.
- [22] J. Goutsias, "Mutually compatible Gibbs random fields," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1233–1249, June 1989.
- [23] D. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Mach. Learn.*, vol. 20, pp. 197–243, 1995.
- [24] F. Heitz, P. Pérez, and P. Bouthemy, "Multiscale minimization of global energy functions in some visual recovery problems," *CVGIP: Image Understand.*, vol. 59, no. 1, pp. 125–134, 1994.
- [25] W. Irving, P. Fieguth, and A. Willisky, "An overlapping tree approach to multiscale stochastic modeling and estimation," *IEEE Trans. Image Processing*, vol. 6, pp. 1517–1529, Nov. 1997.
- [26] F. Jensen, *An Introduction to Bayesian Networks*. London, U.K.: Univ. College London Press, 1996.
- [27] F. Jensen, S. Lauritzen, and K. Olesen, "Bayesian updating in recursive graphical models by local computations," *Comput. Stat. Quart.*, vol. 4, pp. 269–282, 1990.
- [28] A. Kannan, M. Ostendorf, W. Karl, D. Castanon, and R. Fish, "ML parameter estimation of a multiscale stochastic process using the EM algorithm," Boston Univ., Boston, MA, Tech. Rep. ECE-96-009, Nov. 1996.
- [29] Z. Kato, M. Berthod, and Z. Zerubia, "A hierarchical Markov random field model and multi-temperature annealing for parallel image classification," *Graph. Mod. Image Process.*, vol. 58, no. 1, pp. 18–37, 1996.
- [30] S. Krishnamachari and R. Chellappa, "Multiresolution Gauss-Markov random field models for texture segmentation," *IEEE Trans. Image Processing*, vol. 6, pp. 251–267, Feb. 1997.
- [31] J.-M. Laferté, F. Heitz, P. Pérez, and E. Fabre, "Hierarchical statistical models for the fusion of multiresolution image data," in *Proc. Int. Conf. Computer Vision*, June 1995.
- [32] S. Lakshmanan and H. Derin, "Gaussian Markov random fields at multiple resolutions," in *Markov Random Fields: Theory and Applications*, R. Chellappa and A. K. Jain, Eds. New York: Academic, 1993, pp. 131–157.
- [33] S. Lauritzen, *Graphical Models*. New York: Oxford, 1996.
- [34] M. Luetctgen, W. Karl, and A. Willisky, "Efficient multiscale regularization with applications to the computation of optical flow," *IEEE Trans. Image Processing*, vol. 3, pp. 41–64, Jan. 1994.
- [35] M. Luetctgen, W. Karl, A. Willisky, and R. Tenney, "Multiscale representation of Markov random fields," *IEEE Trans. Signal Processing*, vol. 41, pp. 3377–3396, Dec. 1993.
- [36] M. Luetctgen and A. Willisky, "Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination," *IEEE Trans. Image Processing*, vol. 4, pp. 194–207, Feb. 1995.
- [37] J. Moura and N. Balram, "Recursive structure of noncausal Gauss-Markov random fields," *IEEE Trans. Inform. Theory*, vol. 38, pp. 335–354, Feb. 1992.
- [38] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 360–378, Oct. 1996.
- [39] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
- [40] D. Pickard, "A curious binary lattice," *J. Appl. Prob.*, vol. 14, pp. 717–731, 1977.
- [41] P. Pérez, A. Chardin, and J.-M. Laferté, "Noniterative manipulation of discrete energy-based models for image analysis," *Pattern Recognit.*, vol. 33, pp. 573–586, Apr. 2000, to be published.
- [42] P. Pérez and F. Heitz, "Restriction of a Markov random field on a graph and multiresolution statistical image modeling," *IEEE Trans. Inform. Theory*, vol. 42, pp. 180–190, Jan. 1996.
- [43] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA J.*, vol. 3, pp. 1445–1450, 1965.
- [44] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, 1984.
- [45] P. Smyth, D. Heckerman, and M. Jordan, "Probabilistic independence networks for hidden Markov probability models," *Neural Comput.*, vol. 9, no. 2, pp. 227–269, 1997.
- [46] D. Tretter, C. Bouman, K. Khawaja, and A. Maciejewski, "A multiscale stochastic image model for automated inspection," *IEEE Trans. Image Processing*, vol. 4, pp. 1641–1654, Dec. 1995.
- [47] G. Wei and M. Tanner, "A Monte-Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *J. Amer. Stat. Assoc.*, vol. 85, pp. 699–704, 1990.
- [48] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. New York: Wiley, 1990.
- [49] J. Zhang, J. Modestino, and D. Langan, "Maximum-likelihood parameter estimation for unsupervised stochastic model-based image segmentation," *IEEE Trans. Image Processing*, vol. 3, pp. 404–419, Apr. 1994.

Jean-Marc Laferté was born in 1968. He received the Ph.D. degree in computer science from the University of Rennes, Rennes, France, in 1996.

He is now an Assistant Professor with the Computer Science Department, University of Rennes. His research interests include statistical models in image analysis.

Patrick Pérez was born in 1968. He graduated from École Centrale Paris, Paris, France, in 1990. He received the Ph.D. degree in signal processing and telecommunications from the University of Rennes, Rennes, France, in 1993.

He now holds a fulltime research position at the Inria Center, Rennes. His research interests include statistical and/or hierarchical models for large inverse problems in image analysis.

Fabrice Heitz received the engineer degree in electrical engineering and telecommunications from Telecom Bretagne, Bretagne, France, in 1984 and the Ph.D. degree from Telecom Paris, Paris, France, in 1988.

From 1988 until 1994, he was with INRIA Rennes, Rennes, France, as a Senior Researcher in image processing and computer vision. He is now a Professor at Ecole Nationale Supérieure de Physique, Laboratoire des Sciences de l'Image, de l'Informatique et de la Teledetection, Strasbourg, France (ENSPS/LSIIT). His research interests include statistical image modeling, image sequence analysis and medical image analysis.

Dr. Heitz is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.