PROCEEDINGS OF THE EUSIPCO 2004, VIENNA, AUSTRIA, SEPTEMBER 2004

MULTISCALE BAYESIAN ESTIMATION IN PAIRWISE MARKOV TREES

François Desbouvries and Jean Lecomte

GET / INT / dépt. CITI and CNRS UMR 5157
9 rue Charles Fourier, 91011 Evry, France
Francois.Desbouvries@int-evry.fr

ABSTRACT

An important problem in multiresolution analysis of signals and images consists in estimating hidden random variables (r.v.) $\mathbf{x} = \{\mathbf{x}_s\}_{s\in\mathscr{S}}$ from observed ones $\mathbf{y} = \{\mathbf{y}_s\}_{s\in\mathscr{S}}$. This is done classically in the context of Hidden Markov Trees (HMT). In particular, a smoothing Kalman-like algorithm has been proposed by Chou *et al.* in the linear Gaussian case. In this paper we extend this algorithm to the more general framework of Pairwise Markov Trees (PMT).

1. INTRODUCTION

Multiresolution analysis and multiscale algorithms are of interest in a large variety of signal and image processing problems (see e.g. [1] and the references therein). Efficient restoration algorithms have been developed in the context of tree-based structures [2] [3] [4]. These algorithms estimate the hidden r.v. $\mathbf{x}$ from the observed ones $\mathbf{y}$, under the assumption that the stochastic interactions of $\mathbf{x}$ and $\mathbf{y}$ are modeled by an HMT.

On the other hand, it is well known that if $(\mathbf{x},\mathbf{y})$ is a classical Hidden Markov Model (HMM), then the pair $(\mathbf{x},\mathbf{y})$ itself is Markovian. Conversely, starting from the sole assumption that $(\mathbf{x},\mathbf{y})$ is Markovian, i.e. that $(\mathbf{x},\mathbf{y})$ is a so-called Pairwise Markov Model (PMM), is a more general point of view which nevertheless enables the development of similar restoration algorithms. More precisely, some of the classical Bayesian restoration algorithms used in Hidden Markov Fields (HMF), Hidden Markov Chains (HMC) or Hidden Markov Trees (HMT), have been generalized recently to the more general frameworks of Pairwise Markov Fields (PMF) [5], Pairwise Markov Chains (PMC) with discrete [6] or continuous [7] [8] state process, and of PMT with discrete [9] [10] or continuous [10] hidden variables.

In this paper we focus on the smoothing Kalman-like algorithm of Chou *et al.* [3]. This algorithm is an extension to HMT of the RTS smoother [11] derived in the HMC framework, and was later on recognized as being a particular case of Pearl's belief propagation algorithm for Directed Acyclic Graphs. The main aim of this paper is to extend the algorithm of [3] in another direction, i.e. from HMT to PMT models.

This paper is organized as follows. In section 2 we briefly recall the HMT and PMT models, and show that PMT are more general than HMT. A general overview of our extension to the PMT model of the algorithm of [3] is given in section 3.1, and computational details of our algorithm are derived in sections 3.2 and 3.3.

2. HIDDEN VS. PAIRWISE MARKOV TREES

Let $\mathscr{S}$ be a finite set of indices, and let us consider a tree stucture with nodes indexed on $\mathscr{S}$. Let us partition $\mathscr{S}$ in terms of its successive generations $\mathscr{S}_1 \cdots, \mathscr{S}_n$. So, $\mathscr{S}_1$ is made of the root node r, $\mathscr{S}_2$ gathers the children of node r, and so on. Each node s (apart from the root node r) has one father s^-. The set of all descendants of a node s is denoted by s^{++}. We assume for notational simplicity that the tree is dyadic, i.e. that each node s (which is not in the last generation n) has exactly two children s_1 and s_2.

Let now $\mathbf{x} = \{\mathbf{x}_s\}_{s\in\mathscr{S}}$ and $\mathbf{y} = \{\mathbf{y}_s\}_{s\in\mathscr{S}}$ be two sets of r.v. indexed on $\mathscr{S}$. Each $\mathbf{x}_s$ (resp. $\mathbf{y}_s$) belongs to $\mathbb{R}^p$ (resp. to $\mathbb{R}^q$). Let $p(\mathbf{x}_s)$ (resp. $p(\mathbf{y}_s)$) denote the probability density function (p.d.f.) of $\mathbf{x}_s$ (resp. of $\mathbf{y}_s$) w.r.t. Lebesgue measure, and let $p(\mathbf{x}_s|\{\mathbf{y}_\sigma\}_{\sigma\in\Sigma})$ denote the conditional p.d.f. of $\mathbf{x}_s$ given $\{\mathbf{y}_\sigma\}_{\sigma\in\Sigma}$. Other p.d.f. or conditional p.d.f. of interest are defined similarly.

The classical HMT model is widely used for modeling $p(\mathbf{x},\mathbf{y})$. In this model, $\mathbf{x}$ is a Markov Tree (MT), and conditionally on $\mathbf{x}$, the variables $\mathbf{y}_s$ are independent and satisfy $p(\mathbf{y}_s|\mathbf{x}) = p(\mathbf{y}_s|\mathbf{x}_s)$:

$$p(\mathbf{x},\mathbf{y}) = \underbrace{p(\mathbf{x}_r)\prod_{i=2}^{n}\prod_{s\in\mathscr{S}_i} p(\mathbf{x}_s|\mathbf{x}_{s^-})}_{p(\mathbf{x})} \times \underbrace{\prod_{s\in\mathscr{S}} p(\mathbf{y}_s|\mathbf{x}_s)}_{p(\mathbf{y}|\mathbf{x})}. \quad (1)$$

Now, let us introduce the pair $\mathbf{z}_s = (\mathbf{x}_s,\mathbf{y}_s)$, and let $\mathbf{z} = \{\mathbf{z}_s\}_{s\in\mathscr{S}}$. A PMT model is a model in which we only assume that $\mathbf{z}$ is a MT :

$$p(\mathbf{z}) = p(\mathbf{z}_r)\prod_{i=2}^{n}\prod_{s\in\mathscr{S}_i} p(\mathbf{z}_s|\mathbf{z}_{s^-}). \quad (2)$$

One can check easily that (1) implies (2), so any HMT is a PMT. However, PMT are more general than HMT, because if (2) holds, $\mathbf{x}$ is not necessarily a MT, as we see from the following result :

Proposition 1 *Let $\mathbf{z}$ be a dyadic PMT satisfying (2). Assume that*

$$\text{For all } s \in \mathscr{S}\setminus\mathscr{S}_1, \quad p(\mathbf{x}_s|\mathbf{x}_{s^-},\mathbf{y}_{s^-}) = p(\mathbf{x}_s|\mathbf{x}_{s^-}). \quad (3)$$

Then $\mathbf{x}$ is a MT. Conversely, assume that $\mathbf{x}$ is a MT, and that for all $s \in \mathscr{S}\setminus\mathscr{S}_n$, $p(\mathbf{z}_{s_1}|\mathbf{z}_s) = p(\mathbf{z}_{s_2}|\mathbf{z}_s)$, i.e. that conditionally on the father, the laws of the children are equal. Then (3) holds.

Proof. Let $\mathbf{z}_{i:j} = \{\mathbf{z}_s\}_{s \in \mathscr{S}_k, i \leq k \leq j}$, and let us define $\mathbf{x}_{i:j}$ and $\mathbf{y}_{i:j}$ similarly. Using (2) and (3), we get

$$p(\mathbf{x}_{1:n}) = \int p(\mathbf{z}_{1:n}) d\mathbf{y}_{1:n} = \int p(\mathbf{z}_{1:n-1}) [\prod_{s \in \mathscr{S}_n} \int p(\mathbf{z}_s|\mathbf{z}_{s^-}) d\mathbf{y}_s] d\mathbf{y}_{1:n-1}$$
$$= p(\mathbf{x}_{1:n-1}) \prod_{s \in \mathscr{S}_n} p(\mathbf{x}_s|\mathbf{x}_{s^-}) = p(\mathbf{x}_r) \prod_{i=2}^{n} \prod_{s \in \mathscr{S}_i} p(\mathbf{x}_s|\mathbf{x}_{s^-}),$$

so $\mathbf{x}$ is a MT. Conversely, let $\mathbf{x}$ and $\mathbf{z}$ be both MT. Then for all $s \in \mathscr{S} \setminus \mathscr{S}_n$,

$$p(\mathbf{z}_s, \mathbf{z}_{s_1}, \mathbf{z}_{s_2}) = \frac{p(\mathbf{z}_s, \mathbf{z}_{s_1}) p(\mathbf{z}_s, \mathbf{z}_{s_2})}{p(\mathbf{z}_s)} =$$

$$\frac{p(\mathbf{y}_s, \mathbf{y}_{s_1}|\mathbf{x}_s, \mathbf{x}_{s_1}) p(\mathbf{y}_s, \mathbf{y}_{s_2}|\mathbf{x}_s, \mathbf{x}_{s_2})}{p(\mathbf{y}_s|\mathbf{x}_s)} \times \underbrace{\frac{p(\mathbf{x}_s, \mathbf{x}_{s_1}) p(\mathbf{x}_s, \mathbf{x}_{s_2})}{p(\mathbf{x}_s)}}_{p(\mathbf{x}_s, \mathbf{x}_{s_1}, \mathbf{x}_{s_2})}.$$

Integrating w.r.t. $\mathbf{y}_s$, $\mathbf{y}_{s_1}$ and $\mathbf{y}_{s_2}$, we get

$$\int \frac{p(\mathbf{y}_s|\mathbf{x}_s, \mathbf{x}_{s_1}) p(\mathbf{y}_s|\mathbf{x}_s, \mathbf{x}_{s_2})}{p(\mathbf{y}_s|\mathbf{x}_s)} d\mathbf{y}_s = 1. \tag{4}$$

Let $p^i_\omega(\mathbf{y}_s) = p(\mathbf{y}_s|\mathbf{x}_s, \mathbf{x}_{s_i} = \omega)$. By assumption,

$$p^1_\omega(\mathbf{y}_s) = p^2_\omega(\mathbf{y}_s). \tag{5}$$

Using (5) and then (4), we get

$$\int (p^1_\omega(\mathbf{y}_s) - p^1_{\omega'}(\mathbf{y}_s))^2 \frac{1}{p(\mathbf{y}_s|\mathbf{x}_s)} d\mathbf{y}_s = \underbrace{\int \frac{p^1_\omega(\mathbf{y}_s) p^2_\omega(\mathbf{y}_s)}{p(\mathbf{y}_s|\mathbf{x}_s)} d\mathbf{y}_s}_{1}$$

$$+ \underbrace{\int \frac{p^1_{\omega'}(\mathbf{y}_s) p^2_{\omega'}(\mathbf{y}_s)}{p(\mathbf{y}_s|\mathbf{x}_s)} d\mathbf{y}_s}_{1} - 2 \underbrace{\int \frac{p^1_\omega(\mathbf{y}_s) p^2_{\omega'}(\mathbf{y}_s)}{p(\mathbf{y}_s|\mathbf{x}_s)} d\mathbf{y}_s}_{1} = 0.$$

So $p^1_\omega(\mathbf{y}_s) = p^1_{\omega'}(\mathbf{y}_s)$ (and similarly $p^2_\omega(\mathbf{y}_s) = p^2_{\omega'}(\mathbf{y}_s)$), which proves that conditionally on $\mathbf{x}_s$, $\mathbf{x}_{s_i}$ and $\mathbf{y}_s$ are independent. ■

Finally, let us notice that the wider generality of model (2) w.r.t. model (1) is maybe best appreciated at the local level, for the transition p.d.f. $p(\mathbf{z}_s|\mathbf{z}_{s^-})$ in (2) reads

$$\begin{aligned} p(\mathbf{z}_s|\mathbf{z}_{s^-}) &= p(\mathbf{x}_s, \mathbf{y}_s|\mathbf{x}_{s^-}, \mathbf{y}_{s^-}) \\ &= p(\mathbf{x}_s|\mathbf{x}_{s^-}, \mathbf{y}_{s^-}) p(\mathbf{y}_s|\mathbf{x}_s, \mathbf{x}_{s^-}, \mathbf{y}_{s^-}); \end{aligned}$$

so an HMT is a PMT in which $p(\mathbf{x}_s|\mathbf{x}_{s^-}, \mathbf{y}_{s^-})$ reduces to $p(\mathbf{x}_s|\mathbf{x}_{s^-})$, and $p(\mathbf{y}_s|\mathbf{x}_s, \mathbf{x}_{s^-}, \mathbf{y}_{s^-})$ reduces to $p(\mathbf{y}_s|\mathbf{x}_s)$.

3. COMPUTATION OF THE POSTERIOR P.D.F. OF A GIVEN NODE

From now on we shall assume that $\mathbf{z}$ is a Gaussian PMT. The aim of this section consists in computing the posterior p.d.f. $p(\mathbf{x}_s|\mathbf{y})$ for an arbitrary $s \in \mathscr{S}$.

3.1 Modeling assumptions and structure of the algorithm

We assume that (2) holds, and moreover that

$$\mathbf{z}_{s_i} = \mathbf{F}_{s_i} \mathbf{z}_s + \mathbf{w}_{s_i}, \;\; E(\mathbf{w}_{s_i} \mathbf{w}_{s_i}^T) = \mathbf{Q}_{s_i}, \tag{6}$$

in which $\mathbf{w} = \{\mathbf{w}_s\}_{s \in \mathscr{S} \setminus \mathscr{S}_1}$ are random vectors which are zero-mean, independent and independent of $\mathbf{z}_r$, and in which $\mathbf{Q}_s$ is positive definite ($\mathbf{Q}_s > 0$) for all s. We also assume that $\mathbf{w}$ is Gaussian and that $p(\mathbf{z}_r) \sim \mathscr{N}(\mathbf{0}, \mathbf{Q}_r)$. As a consequence, $\mathbf{z}$ is zero-mean and Gaussian and we set $p(\mathbf{z}_s) \sim \mathscr{N}(\mathbf{0}, \mathbf{P}_s)$. All conditionnal p.d.f. related to $\mathbf{z}$ are also Gaussian, so computing these p.d.f. amounts to computing their parameters. Let us thus introduce the following notations :

$$p(\underbrace{\mathbf{x}_s, \mathbf{y}_s}_{\mathbf{z}_s} | \{\mathbf{y}_\sigma\}_{\sigma \in \Sigma}) \;\sim\; \mathscr{N}(\underbrace{\begin{bmatrix} \widehat{\mathbf{x}}_{s|\Sigma} \\ \widehat{\mathbf{y}}_{s|\Sigma} \end{bmatrix}}_{\widehat{\mathbf{z}}_{s|\Sigma}}, \underbrace{\begin{bmatrix} \mathbf{P}^{x,x}_{s|\Sigma} & \mathbf{P}^{x,y}_{s|\Sigma} \\ \mathbf{P}^{y,x}_{s|\Sigma} & \mathbf{P}^{y,y}_{s|\Sigma} \end{bmatrix}}_{\mathbf{P}_{s|\Sigma}}). \tag{7}$$

Following [3], our algorithm is essentially made of two sweeps, one filtering sweep in the backward (fine-to-coarse) direction and then one smoothing sweep in the forward (coarse-to-fine) direction. More precisely, the structure of the algorithm is a follows :

1. From $p(\mathbf{z}_r)$ and equation (6), we compute recursively $p(\mathbf{z}_s)$ for all $s \in \mathscr{S}$ via
$$\mathbf{P}_{s_i} = \mathbf{Q}_{s_i} + \mathbf{F}_{s_i} \mathbf{P}_s \mathbf{F}_{s_i}^T; \tag{8}$$
2. *Fine-to-coarse sweep.* Starting from $\{p(\mathbf{x}_s|\mathbf{y}_s)\}_{s \in \mathscr{S}_n}$, we compute recursively, in the fine-to-coarse direction, $\{p(\mathbf{x}_s|\mathbf{y}_s, \mathbf{y}_{s^{++}})\}_{s \in \mathscr{S}_m}$ for all $m \in \{n-1, \cdots, 1\}$. Since each p.d.f. $p(\mathbf{x}_s|\mathbf{y}_s, \mathbf{y}_{s^{++}})$ is computed from $\{p(\mathbf{x}_{s_i}|\mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})\}_{i=1}^2$ (see §3.2 for details), the computations of the 2^{m-1} p.d.f. $p(\mathbf{x}_s|\mathbf{y}_s, \mathbf{y}_{s^{++}})$ of a given generation m can be performed in parallel. At the end of this backward sweep, $p(\mathbf{x}_r|\mathbf{y})$ has been computed;
3. *Coarse-to-fine sweep.* It remains to compute $p(\mathbf{x}_s|\mathbf{y})$ for an arbitrary s. There is a unique path $\{\sigma_i\}_{i=1}^m$ (with $\sigma_1 = r$ and $\sigma_m = s$) relating node s to the root node r. Along this path, the conditional law of $\{\mathbf{x}_{\sigma_i}\}_{i=1}^m$ given $\mathbf{y}$ is Markovian, so $p(\mathbf{x}_s|\mathbf{y})$ can be computed recursively from $p(\mathbf{x}_{\sigma_1}|\mathbf{y})$ and $\{p(\mathbf{x}_{\sigma_i}|\mathbf{x}_{(\sigma_i)^-}, \mathbf{y})\}_{i=2}^m$. On the other hand, we will see in section 3.3 that each p.d.f. $p(\mathbf{x}_{s_i}|\mathbf{x}_s, \mathbf{y})$ can be computed from $p(\mathbf{x}_s|\mathbf{y})$, and from $p(\mathbf{x}_{s_i}|\mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$ and $p(\mathbf{z}_s|\mathbf{z}_{s_i})$ which have been computed previously.

We now turn to the computational details of the algorithm. The backward sweep is explained in section 3.2 and the forward sweep in section 3.3. The derivations rely on two ingredients. Firstly, the PMT assumption plays an important role; in particular, the following two properties of Markov trees will prove useful in the sequel :

- (P1). Let $s \in \mathscr{S}_m$ with $1 < m < n$. Conditionally on $\mathbf{z}_s$, $\{\mathbf{z}_s^{++}\}$ and $\{\mathbf{z}_\sigma\}_{\mathscr{S} \setminus \{s, s^{++}\}}$ are independent.
- (P2). Let $s \in \mathscr{S}_m$ with $1 \leq m < n$. Conditionally on $\mathbf{z}_s$, $\{\mathbf{z}_{s_1}, \mathbf{z}_{s_1^{++}}\}$ and $\{\mathbf{z}_{s_2}, \mathbf{z}_{s_2^{++}}\}$ are independent.

Secondly, the algorithm also heavily relies on the Gaussian assumption; in particular, we extensively use Propositions 6 and 7 (see the Annex), which is arguably simpler than the approach of [3].

3.2 Fine-to-coarse sweep

Each elementary step of the backward sweep can be decomposed into 3 substeps :

1. *backward prediction step* : for $i = 1,2$, computation of $p(\mathbf{z}_s|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}})$ from $p(\mathbf{x}_{s_i}|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}})$;
2. *fusion step* : computation of $p(\mathbf{z}_s|\mathbf{y}_{s^{++}})$ from $p(\mathbf{z}_s|\mathbf{y}_{s_1},\mathbf{y}_{s_1^{++}})$ and $p(\mathbf{z}_s|\mathbf{y}_{s_2},\mathbf{y}_{s_2^{++}})$;
3. *measurement-update step* : computation of $p(\mathbf{x}_s|\mathbf{y}_s,\mathbf{y}_{s^{++}})$ from $p(\mathbf{z}_s|\mathbf{y}_{s^{++}})$.

These three substeps are described respectively by the following three propositions :

Proposition 2 (Backward prediction step.) *$p(\mathbf{z}_s|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}})$ can be computed from $p(\mathbf{x}_{s_i}|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}})$ via the following recursion :*

$$\widehat{\mathbf{z}}_{s|s_i,s_i^{++}} = \widetilde{\mathbf{F}}_{s_i}\begin{bmatrix}\widehat{\mathbf{x}}_{s_i|s_i,s_i^{++}}\\ \mathbf{y}_{s_i}\end{bmatrix}, \tag{9}$$

$$\mathbf{P}_{s|s_i,s_i^{++}} = \widetilde{\mathbf{Q}}_{s_i}+\begin{bmatrix}\widetilde{\mathbf{F}}^{x,x}_{s_i}\\ \widetilde{\mathbf{F}}^{y,x}_{s_i}\end{bmatrix}\mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}}\left[(\widetilde{\mathbf{F}}^{x,x}_{s_i})^T(\widetilde{\mathbf{F}}^{y,x}_{s_i})^T\right], \tag{10}$$

$$\widetilde{\mathbf{F}}_{s_i} = \mathbf{P}_s\mathbf{F}^T_{s_i}\mathbf{P}^{-1}_{s_i}, \tag{11}$$

$$\widetilde{\mathbf{Q}}_{s_i} = \mathbf{P}_s-\mathbf{P}_s\mathbf{F}^T_{s_i}\mathbf{P}^{-1}_{s_i}\mathbf{F}_{s_i}\mathbf{P}_s. \tag{12}$$

Proof. From (P1), we have for $i = 1,2$,

$$\begin{aligned}p(\mathbf{x}_{s_i},\mathbf{z}_s|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}}) &= p(\mathbf{x}_{s_i}|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}})p(\mathbf{z}_s|\mathbf{z}_{s_i},\mathbf{y}_{s_i^{++}})\\ &\overset{(P1)}{=} p(\mathbf{x}_{s_i}|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}})p(\mathbf{z}_s|\mathbf{z}_{s_i}). \end{aligned}\tag{13}$$

We first need to compute $p(\mathbf{z}_s|\mathbf{z}_{s_i})$ from $p(\mathbf{z}_s)$ and $p(\mathbf{z}_{s_i}|\mathbf{z}_s)$. Using Proposition 7, (8) and Proposition 6, we get

$$p(\mathbf{z}_s,\mathbf{z}_{s_i}) = \underbrace{p(\mathbf{z}_s)}_{\mathcal{N}(\mathbf{0},\mathbf{P}_s)}\underbrace{p(\mathbf{z}_{s_i}|\mathbf{z}_s)}_{\mathcal{N}(\mathbf{F}_{s_i}\mathbf{z}_s,\mathbf{Q}_{s_i})} \sim \mathcal{N}(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix},\begin{bmatrix}\mathbf{P}_s & \mathbf{P}_s\mathbf{F}^T_{s_i}\\ \mathbf{F}_{s_i}\mathbf{P}_s & \mathbf{P}_{s_i}\end{bmatrix}), \tag{14}$$

and so $p(\mathbf{z}_s|\mathbf{z}_{s_i}) \sim \mathcal{N}(\widetilde{\mathbf{F}}_{s_i}\mathbf{z}_{s_i},\widetilde{\mathbf{Q}}_{s_i})$, in which $\widetilde{\mathbf{F}}_{s_i}$ and $\widetilde{\mathbf{Q}}_{s_i}$ are given by (11) and (12). We next turn back to the computation of (13). Using Proposition 7, we have

$$\begin{aligned}p(\mathbf{x}_{s_i},\mathbf{z}_s|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}}) = &\underbrace{p(\mathbf{x}_{s_i}|\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}})}_{\mathcal{N}(\widehat{\mathbf{x}}_{s_i|s_i,s_i^{++}},\mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}})}\ \underbrace{p(\mathbf{z}_s|\mathbf{x}_{s_i},\mathbf{y}_{s_i})}_{\mathcal{N}(\widetilde{\mathbf{F}}_{s_i}\mathbf{z}_{s_i},\widetilde{\mathbf{Q}}_{s_i})}\\ &\sim\mathcal{N}(\begin{bmatrix}\widehat{\mathbf{x}}_{s_i|s_i,s_i^{++}}\\ \widetilde{\mathbf{F}}_{s_i}\begin{bmatrix}\widehat{\mathbf{x}}_{s_i|s_i,s_i^{++}}\\ \mathbf{y}_{s_i}\end{bmatrix}\end{bmatrix},\\ &\begin{bmatrix}\mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}} & \mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}}\left[(\widetilde{\mathbf{F}}^{x,x}_{s_i})^T(\widetilde{\mathbf{F}}^{y,x}_{s_i})^T\right]\\ \begin{bmatrix}\widetilde{\mathbf{F}}^{x,x}_{s_i}\\ \widetilde{\mathbf{F}}^{y,x}_{s_i}\end{bmatrix}\mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}} & \widetilde{\mathbf{Q}}_{s_i}+\begin{bmatrix}\widetilde{\mathbf{F}}^{x,x}_{s_i}\\ \widetilde{\mathbf{F}}^{y,x}_{s_i}\end{bmatrix}\mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}}\left[(\widetilde{\mathbf{F}}^{x,x}_{s_i})^T(\widetilde{\mathbf{F}}^{y,x}_{s_i})^T\right]\end{bmatrix}),\end{aligned}\tag{15}$$

from which we deduce (9) and (10). ■

Proposition 3 (Fusion step.) *$p(\mathbf{z}_s|\mathbf{y}_{s^{++}})$ can be computed from $p(\mathbf{z}_s|\mathbf{y}_{s_1},\mathbf{y}_{s_1^{++}})$ and $p(\mathbf{z}_s|\mathbf{y}_{s_2},\mathbf{y}_{s_2^{++}})$ via :*

$$\widehat{\mathbf{z}}_{s|s^{++}} = \mathbf{P}_{s|s^{++}}[\sum_{i=1}^{2}\mathbf{P}^{-1}_{s|s_i,s_i^{++}}\widehat{\mathbf{z}}_{s|s_i,s_i^{++}}], \tag{16}$$

$$\mathbf{P}_{s|s^{++}} = [\,\mathbf{P}^{-1}_{s|s_1,s_1^{++}}+\mathbf{P}^{-1}_{s|s_2,s_2^{++}}-\mathbf{P}^{-1}_s\,]^{-1}. \tag{17}$$

Proof. We are going to compute $p(\mathbf{z}_s|\mathbf{y}_{s^{++}})$ from $p(\mathbf{z}_s,\mathbf{y}_{s^{++}}) = p(\mathbf{z}_s)\ p(\mathbf{y}_{s^{++}}|\mathbf{z}_s)$. From (P2), $p(\mathbf{y}_{s^{++}}|\mathbf{z}_s) = p(\mathbf{y}_{s_1},\mathbf{y}_{s_1^{++}}|\mathbf{z}_s)p(\mathbf{y}_{s_2},\mathbf{y}_{s_2^{++}}|\mathbf{z}_s)$. On the other hand, due to (6),

$$\begin{bmatrix}\mathbf{y}_{s_i}\\ \mathbf{y}_{s_i^{++}}\end{bmatrix} = \mathbf{A}_{s_i}\mathbf{z}_s+\mathbf{B}_{s_i}\begin{bmatrix}\mathbf{w}_{s_i}\\ \mathbf{w}_{s_i^{++}}\end{bmatrix} \tag{18}$$

for some matrices $\mathbf{A}_{s_i}$ and $\mathbf{B}_{s_i}$. Let $\mathbf{\Pi}_{s_i} = \mathrm{Cov}(\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}}|\mathbf{z}_s)$. Since $\mathbf{z}_s$ and $[\mathbf{w}^T_{s_i},\mathbf{w}^T_{s_i^{++}}]^T$ are independent, (18) yields $p(\mathbf{y}_{s_i},\mathbf{y}_{s_i^{++}}|\mathbf{z}_s) \sim \mathcal{N}(\mathbf{A}_{s_i}\mathbf{z}_s,\mathbf{\Pi}_{s_i})$. Next, using Proposition 7, we get

$$\begin{aligned}p(\mathbf{z}_s,\mathbf{y}_{s^{++}}) = &\underbrace{p(\mathbf{z}_s)}_{\mathcal{N}(\mathbf{0},\mathbf{P}_s)}\times\underbrace{p(\mathbf{y}_{s_1},\mathbf{y}_{s_1^{++}},\mathbf{y}_{s_2},\mathbf{y}_{s_2^{++}}|\mathbf{z}_s)}_{\mathcal{N}(\begin{bmatrix}\mathbf{A}_{s_1}\mathbf{z}_s\\ \mathbf{A}_{s_2}\mathbf{z}_s\end{bmatrix},\begin{bmatrix}\mathbf{\Pi}_{s_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Pi}_{s_2}\end{bmatrix})}\\ \sim\mathcal{N}(\mathbf{0},&\begin{bmatrix}\mathbf{P}_s & \mathbf{P}_s[\mathbf{A}^T_{s_1}\mathbf{A}^T_{s_2}]\\ \begin{bmatrix}\mathbf{A}_{s_1}\\ \mathbf{A}_{s_2}\end{bmatrix}\mathbf{P}_s & \begin{bmatrix}\mathbf{\Pi}_{s_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Pi}_{s_2}\end{bmatrix}+\begin{bmatrix}\mathbf{A}_{s_1}\\ \mathbf{A}_{s_2}\end{bmatrix}\mathbf{P}_s[\mathbf{A}^T_{s_1}\mathbf{A}^T_{s_2}]\end{bmatrix}).\end{aligned}\tag{19}$$

We can now compute $p(\mathbf{z}_s|\mathbf{y}_{s^{++}})$ with the help of Proposition 6. Using the matrix inversion lemma, we get

$$\mathbf{P}_{s|s^{++}} = [\mathbf{P}^{-1}_s+\sum_{i=1}^{2}\mathbf{A}^T_{s_i}\mathbf{\Pi}^{-1}_{s_i}\mathbf{A}_{s_i}]^{-1}, \tag{20}$$

$$\mathbf{P}_{s|s_i,s_i^{++}} = [\mathbf{P}^{-1}_s+\mathbf{A}^T_{s_i}\mathbf{\Pi}^{-1}_{s_i}\mathbf{A}_{s_i}]^{-1}, \tag{21}$$

from which we deduce (17). On the other hand, using (20), (21), as well as the well known identity $(\mathbf{A}+\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} = \mathbf{A}^{-1}\mathbf{B}(\mathbf{D}+\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}$, we get

$$\widehat{\mathbf{z}}_{s|s^{++}} = \mathbf{P}_{s|s^{++}}(\sum_{i=1}^{2}\mathbf{A}^T_{s_i}\mathbf{\Pi}^{-1}_{s_i}\begin{bmatrix}\mathbf{y}_{s_i}\\ \mathbf{y}_{s_i^{++}}\end{bmatrix}), \tag{22}$$

$$\widehat{\mathbf{z}}_{s|s_i,s_i^{++}} = \mathbf{P}_{s|s_i,s_i^{++}}\mathbf{A}^T_{s_i}\mathbf{\Pi}^{-1}_{s_i}\begin{bmatrix}\mathbf{y}_{s_i}\\ \mathbf{y}_{s_i^{++}}\end{bmatrix}, \tag{23}$$

from which we deduce (16). ■

Proposition 4 (Measurement-update step.) *$p(\mathbf{x}_s|\mathbf{y}_s,\mathbf{y}_{s^{++}})$ can be computed from $p(\mathbf{z}_s|\mathbf{y}_{s^{++}})$ via the following recursions :*

$$\widehat{\mathbf{x}}_{s|s,s^{++}} = \widehat{\mathbf{x}}_{s|s^{++}}+\mathbf{P}^{x,y}_{s|s^{++}}(\mathbf{P}^{y,y}_{s|s^{++}})^{-1}(\mathbf{y}_s-\widehat{\mathbf{y}}_{s|s^{++}}) \tag{24}$$

$$\mathbf{P}^{x,x}_{s|s,s^{++}} = \mathbf{P}^{x,x}_{s|s^{++}}-\mathbf{P}^{x,y}_{s|s^{++}}(\mathbf{P}^{y,y}_{s|s^{++}})^{-1}\mathbf{P}^{y,x}_{s|s^{++}}. \tag{25}$$

Proof. Use Proposition 6. ■

3.3 Coarse-to-fine sweep

Remember from § 3.1 that the key point of the coarse-to-fine sweep is the recursive computation of $p(\mathbf{x}_{s_i}|\mathbf{y})$ from $p(\mathbf{x}_s|\mathbf{y})$.

Proposition 5 *$p(\mathbf{x}_{s_i}|\mathbf{y})$ can be computed from $p(\mathbf{x}_s|\mathbf{y})$ via the following recursions :*

$$\mathbf{J}_{s_i} = \mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}}\left[(\widetilde{\mathbf{F}}^{x,x}_{s_i})^T(\widetilde{\mathbf{F}}^{y,x}_{s_i})^T\right]\mathbf{P}^{-1}_{s|s_i,s_i^{++}}, \tag{26}$$

$$\widehat{\mathbf{x}}_{s_i|S} = \widehat{\mathbf{x}}_{s_i|s_i,s_i^{++}}+\mathbf{J}_{s_i}(\begin{bmatrix}\widehat{\mathbf{x}}_{s|S}\\ \mathbf{y}_s\end{bmatrix}-\widehat{\mathbf{z}}_{s|s_i,s_i^{++}}), \tag{27}$$

$$\mathbf{P}^{x,x}_{s_i|S} = \mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}}+\mathbf{J}_{s_i}(\begin{bmatrix}\mathbf{P}^{x,x}_{s|S} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}-\mathbf{P}_{s|s_i,s_i^{++}})\mathbf{J}^T_{s_i}. \tag{28}$$

Proof. We are going to compute $p(\mathbf{x}_{s_i}|\mathbf{y})$ via

$$p(\mathbf{x}_{s_i}|\mathbf{y}) = \int p(\mathbf{x}_s|\mathbf{y}) p(\mathbf{x}_{s_i}|\mathbf{x}_s, \mathbf{y}) d\mathbf{x}_s. \qquad (29)$$

Now, from (P1) and (P2) one can show easily that $p(\mathbf{x}_{s_i}|\mathbf{x}_s, \mathbf{y}) = p(\mathbf{x}_{s_i}|\mathbf{x}_s, \mathbf{y}_s, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$. On the other hand, $p(\mathbf{x}_{s_i}, \mathbf{z}_s|\mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$ has already been computed (see eq. (15)). Using Proposition 6, (9) and (10), we see that

$$p(\mathbf{x}_{s_i}|\mathbf{z}_s, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}) \sim \mathscr{N}(\widehat{\mathbf{x}}_{s_i|s_i,s_i^{++}} + \mathbf{J}_{s_i}(\mathbf{z}_s - \widehat{\mathbf{z}}_{s|s_i,s_i^{++}}), \mathbf{C}_{s_i}), \qquad (30)$$

in which $\mathbf{J}_{s_i}$ is given by (26), and $\mathbf{C}_{s_i} = \mathbf{P}^{x,x}_{s_i|s_i,s_i^{++}} - \mathbf{J}_{s_i} \mathbf{P}_{s|s_i,s_i^{++}} \mathbf{J}^T_{s_i}$. So

$$p(\mathbf{x}_{s_i}|\mathbf{x}_s, \mathbf{y}_s, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}) \sim \mathscr{N}(\mathscr{A}_{s_i}\mathbf{x}_s + \mathbf{b}_{s_i}, \mathbf{C}_{s_i})$$

for some matrices $\mathscr{A}_{s_i}$ and $\mathbf{b}_{s_i}$. Coming back to (29), we see from Proposition 7 that

$$\begin{aligned} p(\mathbf{x}_{s_i}|\mathbf{y}) &= \int \underbrace{p(\mathbf{x}_s|\mathbf{y})}_{\mathscr{N}(\widehat{\mathbf{x}}_{s|S}, \mathbf{P}^{x,x}_{s|S})} \underbrace{p(\mathbf{x}_{s_i}|\mathbf{x}_s, \mathbf{y}_s, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})}_{\mathscr{N}(\mathscr{A}_{s_i}\mathbf{x}_s + \mathbf{b}_{s_i}, \mathbf{C}_{s_i})} d\mathbf{x}_s \\ &\sim \mathscr{N}(\mathscr{A}_{s_i}\widehat{\mathbf{x}}_{s|S} + \mathbf{b}_{s_i}, \mathbf{C}_{s_i} + \mathscr{A}_{s_i}\mathbf{P}^{x,x}_{s|S}\mathscr{A}^T_{s_i}), \end{aligned}$$

whence (27) and (28). ■

3.4 Comments and remarks

- The assumption $\mathbf{Q}_s > \mathbf{0}$ in section 3.1 is a simple sufficient condition ensuring that all computations are valid. For if $\mathbf{Q}_s > \mathbf{0}$ for all s, then from (8) we get $\mathbf{P}_s > \mathbf{0}$, so the covariance matrix of $(\mathbf{z}_s, \mathbf{z}_{s_i})$ in (14) is $> \mathbf{0}$ as well, and $\widetilde{\mathbf{Q}}_{s_i}$ in (12) is $> \mathbf{0}$. On the other hand, from (18) we get $\mathbf{\Pi}_s > \mathbf{0}$, so the covariance matrix in (19) is also $> \mathbf{0}$, which in turn ensures that $\mathbf{P}_{s|s_i,s_i^{++}} > \mathbf{0}$, $\mathbf{P}_{s|s^{++}} > \mathbf{0}$ and $\mathbf{P}^{x,x}_{s|s,s^{++}} > \mathbf{0}$. Next, since $\widetilde{\mathbf{Q}}_{s_i} > \mathbf{0}$ and $\mathbf{P}^{x,x}_{s|s,s^{++}} > \mathbf{0}$, the covariance matrix in (15) is $> \mathbf{0}$, so the matrix $\mathbf{C}_{s_i}$ in (30) is $> \mathbf{0}$. Finally $\mathbf{P}^{x,x}_{r|r,r^{++}} = \mathbf{P}^{x,x}_{r|\mathscr{S}} > \mathbf{0}$, and by induction we see that $\mathbf{P}^{x,x}_{s|\mathscr{S}} > \mathbf{0}$ for all s.
- Equations (9) to (28) still hold if we only assume that $\mathbf{P}_s > \mathbf{0}$ for all s (with $\mathbf{Q}_s$ and $\mathbf{\Pi_s}$ possibly singular). The proof is slightly more technical and is omitted here.
- Our algorithm inherits good properties of that of Chou *et al.*. In particular, its complexity is linear in the number of nodes, and its regular pyramidal structure (which is consistent with that of the dyadic tree) yields considerable parallelism in the computations.
- The algorithm can easily be adapted to the case where each node s admits an arbitrary number of children ν_s; of course, depending on the specific tree structure, parallelism may no longer be ensured. All equations remain valid, apart from Proposition 3 which needs to be adapted. The sums in (20) and in (22) run from 1 to ν_s, so (16) and (17) become

$$\begin{aligned} \widehat{\mathbf{z}}_{s|s^{++}} &= \mathbf{P}_{s|s^{++}} \sum_{i=1}^{\nu_s} \mathbf{P}^{-1}_{s|s_i,s_i^{++}} \widehat{\mathbf{z}}_{s|s_i,s_i^{++}}, & (31) \\ \mathbf{P}_{s|s^{++}} &= [\sum_{i=1}^{\nu_s} \mathbf{P}^{-1}_{s|s_i,s_i^{++}} - (\nu_s - 1)\mathbf{P}_s^{-1}]^{-1}. & (32) \end{aligned}$$

A. SOME PROPERTIES OF GAUSSIAN R.V.

The derivations of sections 3.2 and 3.3 rely heavily on the following two properties of Gaussian r.v., which are recalled for convenience of the reader.

Proposition 6 *Let* $p(\mathbf{u}_1, \mathbf{u}_2) \sim \mathscr{N}(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}_{1,1} & \mathbf{\Sigma}_{1,2} \\ \mathbf{\Sigma}_{2,1} & \mathbf{\Sigma}_{2,2} \end{bmatrix})$. *Then* $p(\mathbf{u}_1|\mathbf{u}_2) \sim \mathscr{N}(\mu_{1|2}, \mathbf{\Sigma}_{1|2})$, *with* $\mu_{1|2} = \mu_1 + \mathbf{\Sigma}_{1,2}\mathbf{\Sigma}_{2,2}^{-1}(\mathbf{u}_2 - \mu_2)$ *and* $\mathbf{\Sigma}_{1|2} = \mathbf{\Sigma}_{1,1} - \mathbf{\Sigma}_{1,2}\mathbf{\Sigma}_{2,2}^{-1}\mathbf{\Sigma}_{2,1}$.

Proposition 7 *Let* $p(\mathbf{u}_1) \sim \mathscr{N}(\mu_1, \mathbf{\Sigma}_1)$ *and* $p(\mathbf{u}_2|\mathbf{u}_1) \sim \mathscr{N}(\mathbf{A}\mathbf{u}_1 + \mathbf{b}, \mathbf{\Sigma}_{2|1})$. *Then*

$$p(\mathbf{u}_1, \mathbf{u}_2) \sim \mathscr{N}(\begin{bmatrix} \mu_1 \\ \mathbf{A}\mu_1 + \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{\Sigma}_1\mathbf{A}^T \\ \mathbf{A}\mathbf{\Sigma}_1 & \mathbf{\Sigma}_{2|1} + \mathbf{A}\mathbf{\Sigma}_1\mathbf{A}^T \end{bmatrix}).$$

REFERENCES

[1] A. S. Willsky, “Multiresolution markov models for signal and image processing,” *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1396–1458, August 2002.

[2] C. A. Bouman and M. Shapiro, “A multiscale random field model for bayesian image segmentation,” *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 162–77, March 1994.

[3] K. C. Chou, A. S. Willsky, and A. Benveniste, “Multiscale recursive estimation, data fusion, and regularization,” *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 464–78, March 1994.

[4] J. M. Laferté, P. Perez, and F. Heitz, “Discrete markov image modeling and inference on the quadtree,” *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 390–404, March 2000.

[5] W. Pieczynski and A. N. Tebbache, “Pairwise markov random fields and segmentation of textured images,” *Machine Graphics and Vision*, vol. 9, no. 3, pp. 705–718, March 2000.

[6] W. Pieczynski, “Pairwise markov chains,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 634–639, May 2003.

[7] W. Pieczynski and F. Desbouvries, “Kalman filtering using pairwise gaussian models,” in *Proceedings of the Icassp*, Hong-Kong, April 6-10 2003, vol. 6, pp. VI–57 – VI–60.

[8] F. Desbouvries and W. Pieczynski, “Particle filtering in pairwise and triplet markov chains,” in *Proceedings of the IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP 2003)*, Grado-Gorizia, Italy, June 8-11 2003.

[9] W. Pieczynski, “Arbres de markov couple - pairwise markov trees,” *Comptes Rendus Acad. Sciences - Mathématiques*, vol. 335, pp. 79–82, 2002, Ser. I.

[10] E. Monfrini, J. Lecomte, F. Desbouvries, and W. Pieczynski, “Image and signal restoration using pairwise markov trees,” in *Proceedings of the 2003 IEEE Workshop on Statistical Signal Processing*, St. Louis, MI, September 2003.

[11] H. E. Rauch, F. Tung, and C. T. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA J.*, vol. 3, no. 8, pp. 1445–1450, August 1965.