LIDS-P-2157

Multiscale Representations of Markov Random Fields[1]

Mark R. Luettgen[2]
William C. Karl[3]
Alan S. Willsky[4]
Robert R. Tenney[5]

June 27, 1993

Abstract

Recently, a framework for multiscale stochastic modeling was introduced based on coarse-to-fine scale-recursive dynamics defined on trees. This model class has some attractive characteristics which lead to extremely efficient, statistically optimal signal and image processing algorithms. In this paper, we show that this model class is also quite rich. In particular, we describe how 1-D Markov processes and 2-D Markov random fields (MRF's) can be represented within this framework. The recursive structure of 1-D Markov processes makes them simple to analyze, and generally leads to computationally efficient algorithms for statistical inference. On the other hand, 2-D MRF's are well known to be very difficult to analyze due to their non-causal structure, and thus their use typically leads to computationally intensive algorithms for smoothing and parameter identification. In contrast, our multiscale representations are based on scale-recursive models and thus lead naturally to *scale-recursive* algorithms, which can be substantially more efficient computationally than those associated with MRF models. In 1-D, the multiscale representation is a generalization of the mid-point deflection construction of Brownian motion. The representation of 2-D MRF's is based on a further generalization to a "mid-line" deflection construction. The exact representations of 2-D MRF's are used to motivate a class of multiscale approximate MRF models based on *one-dimensional* wavelet transforms. We demonstrate the use of these latter models in the context of texture representation and, in particular, we show how they can be used as approximations for or alternatives to well-known MRF texture models.

[1] The first three authors were supported by the Draper Laboratory IR&D Program under agreement DL-H-418524, by the Office of Naval Research under Grant N00014-91-J-1004, by the Army Research Office under Grant DAAL03-92-G-0115, and by the Air Force Office of Scientific Research under Grant AFOSR-92-J-0002. The last two authors were also supported by the AFOSR under contract F49620-91-C-0047.

[2] This author was with the M.I.T Laboratory for Information and Decision Systems and is now with Alphatech, Inc., 50 Mall Road, Burlington, MA 01803.

[3] M.I.T Laboratory for Information and Decision Systems, Cambridge, MA 02139.

[4] M.I.T Laboratory for Information and Decision Systems and Department of Electrical Engineering and Computer Science, Cambridge, MA 02139.

[5] Alphatech, Inc., 50 Mall Road, Burlington, MA 01803.

1 Introduction

In this paper, we describe how to use a class of multiscale stochastic models to represent 1-D Markov and reciprocal processes and 2-D Markov random fields (MRF's). Markov models in one dimension provide a rich framework for modeling a wide variety of biological, chemical, electrical, mechanical and economic phenomena [7]. Moreover, the Markov structure makes the models very simple to analyze, so that they often can be easily applied to statistical inference problems (such as detection, parameter identification and state estimation) as well as problems in system design (e.g. control and queuing systems).

In two dimensions, MRF's also have been widely used as models for physical systems [3, 4, 39, 23], and more recently for images. For example, Gaussian fields [45] have been used as image texture models [17, 29, 10, 37], and the more general Gibbs fields have been used as prior models in image segmentation, edge detection and smoothing problems [5, 25, 40, 38]. Causal sub-classes of MRF's, such as Markov Mesh Random Fields [1, 21] and Non-Symmetric Half-Plane Markov chains [28] lead to two-dimensional versions of Kalman filtering algorithms when the fields are Gaussian [46]. In addition, efficient fast Fourier transform algorithms are available for stationary Gaussian fields defined on toroidal lattices [18, 29, 11]. In general, however, Markov random field models lead to computationally intensive algorithms (e.g. stochastic relaxation [25]) for estimation problems. In addition, parameter identification is difficult for MRF models due to the problem of computing the partition function [4, 41]. Thus, while Markov random fields provide a rich structure for multidimensional modeling, they do not generally lead to the simple analysis and computationally efficient algorithms that 1-D Markov processes do.

These computational issues are the most important obstacle to the application of MRF models to a broader range of problems, and are the principal motivations for the investigation in this paper of the richness of the class of multiscale stochastic processes [15, 13, 14, 8, 9], and in particular of how such multiscale processes can be used to exactly and approximately represent Markov random fields. Our multiscale stochastic processes are described by *scale-recursive* models, which lead naturally to computationally efficient *scale-recursive* algorithms for a variety of estimation problems. For instance, fast smoothing algorithms are developed for a class of Gaussian processes in [15, 13, 14]. Also, Bouman and Shapiro demonstrate how a related multiscale discrete random field leads to an efficient sequential MAP estimator [8, 9]. In this paper, we demonstrate how a simple generalization of the models in [15, 13, 14] leads to classes of models which can be used to represent *all* 1-D Markov processes and 2-D Markov random fields. The significance of this result is that it suggests that this multiscale modeling framework may be a decidedly superior basis for image and random field modeling and analysis than the MRF framework both because of the efficient algorithms it admits *and* because of the rich class of phenomena it can be used to describe.

The efficient algorithms to which the multiscale framework leads and which motivate our work here have already led to interesting and substantial new developments in a number of areas. In addition to the work on image segmentation described in [8, 9], in [35] we exploit the multiscale framework to develop new and efficient algorithms for estimating optical flow in image sequences. Standard formulations of this problem require the computationally intensive solution of an elliptic partial differential equation which arises from the often used "smoothness constraint" type regularization, corresponding to regularizing the problem with an MRF prior. We utilize the interpretation of the smoothness constraint as a "fractal prior" to motivate regularization based on one of our multiscale models. The result is a slightly different prior model, which yields comparable root-mean-square (rms) error performance to that achieved using the standard MRF prior model, but with drastically reduced computational load. In particular, in contrast to the iterative algorithms needed to solve the elliptic equations corresponding to the MRF prior, our multiscale algorithm is ***scale-recursive***, yielding the optimal estimate in a finite number of steps with *constant* per pixel computational load. Figure 1, taken from [35], is representative of the results. Here, the rms error is plotted for our multiscale regularization (MR) algorithm and versus iteration for two iterative methods for solving the MRF-based estimation problem. Since the multiscale method is *not* iterative, its rms performance is plotted as a horizontal line. The entire multiscale algorithm has a computational load roughly equal to 4.2 iterations of the iterative successive over-relaxation (SOR) algorithm, indicating a considerable computational savings. Moreover, not only does this computational savings grow with image size (because of the constant per pixel complexity of our approach) but also this algorithm yields error covariance information as part of its computation, something that is not feasible in the smoothness constraint formulation and that can be used to determine the optimal resolution for flow estimation at each point in the image frame (see [35]). Based on this evidence of its promise in practice, it is natural, then, to ask the question of *how rich a class of phenomena can this multiscale formalism capture?* The answer provided in this paper is that this class is extremely rich indeed.

The multiscale representations developed here rely on a generalization of the mid-point deflection technique for constructing a Brownian motion in one dimension [20, 24, 33]. To construct a Brownian motion sample path over an interval by mid-point deflection, we start by randomly choosing values for the process at the mid-point and the two boundary points of the interval according to the joint probability distribution implied by the Brownian motion model. We then use these three values to compute the expected values of the Brownian motion at the one-fourth and three-fourths points of the interval. The expected value at the one-fourth (three-fourths) point corresponds to the average of the initial and mid-point values (mid-point and final values) as shown in the upper left of Figure 2. Random values, with appropriate error variances, are then added to the predictions at each of these new points. The critical observation to be made here is that, since the Brownian motion process is a Markov process, its value at the one-fourth point, given the values at the initial point and mid-point is *independent* of the process values beyond the mid-point, in particular the values at the three-fourths and end-points of the interval. Obviously, it is also the case that the value at the three-fourths point is independent of the values at the initial and one-fourth points, given the values at the mid-point and final point. Consequently, the random deflection terms used to generate the values of the Brownian motion at the one-fourth and three-fourths points can be chosen *independently*. In addition, we see that the Markov property of Brownian motion allows us to iterate this process, generating values at increasingly dense sets of dyadic points in the interval.

There are several important observations to be made about the preceding development. The first is that, by linearly interpolating at each level in this procedure, as illustrated in Figure 2, a sequence of continuous approximations of a Brownian motion is constructed, and the statistics of these approximations converge to those of a Brownian motion [20]. Indeed, this sequence of linear spline approximations can be interpreted exactly as a non-orthogonal multiscale approximation using as the scaling function the triangular "hat" function [42] which is the integral of the Haar wavelet [24]. Second, as we will see, the structure of this mid-point deflection construction fits precisely into the multiscale modeling framework developed in [15, 13, 14], and corresponds simply to a particular choice of the parameters in the multiscale model. Moreover, this concept generalizes, allowing us to show that *all* 1-D reciprocal and Markov processes can be represented by multiscale stochastic models in a similar way. Thus, in one dimension we will show that the class of processes realizable via multiscale, scale-recursive models is at least as rich as the class of all Markov and reciprocal processes. In fact, as we will illustrate, it is significantly richer than this.

Furthermore, these ideas can be extended to multidimensional processes. In particular, we show how a generalization of the mid-point deflection concept to a "mid-line" deflection construction can be used to represent all 2-D MRF's with multiscale models. In particular, the key to our multiscale representations in one or two dimensions is a partitioning of the domain over which the process is defined so that the coarse-to-fine construction of the process can proceed independently in each subdomain. Markovianity plus knowledge of the process on the boundaries of the subdomain partition make this possible. The fundamental difference, however, between the 1-D and 2-D cases is due to the fact that boundaries in $\mathcal{R}^2$ correspond to *curves* or in $\mathcal{Z}^2$ to *sets* of connected lattice sites, as opposed to pairs of points in one dimension. Because of this difference, exact multiscale representations of MRF's defined over a subset of $\mathcal{Z}^2$ have a dimension which varies from scale to scale, and which depends on the size of the domain over which the MRF is defined.

In addition to the exact representations, we will introduce a family of approximate representations for Gaussian MRF's (GMRF's) based on wavelet transforms. As we have indicated, maintaining complete knowledge of a process on 2-D boundaries leads to models of scale-varying dimension, which can become prohibitively large for domains of substantial size. On the other hand, at coarser scales, it would seem reasonable to keep only coarse approximations to these boundary values, and this leads naturally to the use of a multiscale change of basis for the representation of the values of a 2-D process along each 1-D boundary. That is, through our mid-line deflection based models, we are led to the idea of using *one-dimensional wavelet transforms* in the representation of the values of a *two-dimensional* GMRF. The result is a family of models, ranging from those which keep only the coarsest wavelet coefficients along each 1-D boundary to the exact model which keeps them all. This family of approximate representations allows one to tradeoff the complexity and accuracy of the representations, while also providing a framework for the development of extremely efficient estimation and likelihood calculation algorithms. We demonstrate our framework for wavelet-based approximate representation of Gaussian MRF's in the context of natural texture representation [10, 17, 16, 29, 37].

This paper is organized as follows. Section 2 describes the class of multiscale stochastic models that we use. Section 3 develops the details of the representation of Brownian motion discussed above, and generalizes this idea to allow the representation of all 1-D Markov and reciprocal processes. Section 4 then describes how these ideas can be further generalized to provide exact and approximate representations of MRF's. Section 5 illustrates how the approximate models can be used to represent GMRF texture models. In our opinion, one of the conclusions that can be drawn from these results is that this multiscale modeling framework holds substantial promise as an alternative to the MRF framework as it possesses advantages both in terms of the efficient optimal algorithms it leads to *and* in the expressive power it holds. Although a number of interesting and substantive problems remain to be investigated, practical applications of the framework are already emerging and several of these, as well as the conclusions of this paper, are discussed in Section 6.

2 Multiscale Stochastic Models

In this section we describe the classes of multiscale stochastic models to be used in this paper. A class of models for Gaussian processes is described first, followed by a generalization allowing for more general (non-Gaussian) processes. For simplicity, in this section we introduce the basic structure and form of our models in the context of representing 1-D signals and processes. The extension of the models to 2-D is conceptually straightforward, adding only notational and graphical complexity, and thus we defer the introduction of this extension until Section 4, where it is needed.

2.1 Gaussian Multiscale Models

The models presented here and introduced in [15, 13, 14] describe multiscale Gaussian stochastic processes indexed by nodes on the dyadic tree in Figure 3. Different levels of the tree correspond to different scales of the process. In particular, the 2^{m-1} values at the m^{th} level of the tree are interpreted as information about the m^{th} scale of the process, where the notion of "information" at this point is abstract. For instance, values of the process at level m may correspond to averages of pairs of values at level $m+1$. In this case, one can interpret the values of the multiscale process as *scaling coefficients* in a Haar wavelet representation of the process at the finest scale [42]. However, there are many other possible interpretations of the information represented at each level in the tree. For example, values of the multiscale process at a certain level could also correspond to new details of the process not present at coarser resolutions. In this case, the process values would be interpreted as the *wavelet coefficients* in a wavelet representation of a 1-D function or sequence. Alternatively, the values at different levels may correspond to decimated versions of the process at the finest scale. As we will see, this latter interpretation applies to our multiscale representations of reciprocal processes and MRF's, although the representations can also be interpreted in terms of scaling coefficients corresponding to particular non-orthogonal expansions.

We denote nodes on the tree with an abstract index s, and define an upward shift operator $\bar{\gamma}$ such that $s\bar{\gamma}$ is the parent of node s, as illustrated in Figure 3. Also, we define the scale of node s, i.e. the level of the node, as $m(s)$. The stochastic tree process $x_s \in \mathcal{R}^n$ is then described via the following scale-recursive dynamic model:

$$x_s \;=\; A_s x_{s\bar{\gamma}} + B_s w_s \tag{1}$$

under the assumptions[6] $x_0 \sim \mathcal{N}(0, P_0)$ and $w_s \sim \mathcal{N}(0, I)$, where $w_s \in \mathcal{R}^m$ and A_s and B_s are matrices of appropriate size. The state variable x_0 at the root node of the tree provides

[6] The notation $x \sim \mathcal{N}(m, P)$ means the random vector x is normally distributed with mean vector m and covariance matrix P.

an initial condition for the recursion. The driving noise w_s is white and is independent of the initial condition. Interpreting each level as a representation of one scale of the process, we see that (1) describes the evolution of the process from coarse to fine scales. The term $A_s x_{s\bar{\gamma}}$ represents interpolation or prediction down to the next level, and $B_s w_s$ represents new information added as the process evolves from one scale to the next. The choice of the parameters A_s and B_s and their dependence (if any) on the node s, depends upon the particular application and process being modeled [15, 13, 14, 35]. In the context of this paper, as we will see, the parameters of the model (1) are determined in a constructive fashion in order to represent the reciprocal process or MRF of interest.

Note that any given node on the dyadic tree can be viewed as a boundary between three subsets of nodes (two corresponding to paths leading towards offspring and one corresponding to the path leading to a parent). An extremely important property of the scale-recursive model (1) is that not only is it Markov from scale-to-scale, but, conditioned on the value of the state at any node, the values of the states defined at the corresponding three subsets of nodes are independent. This fact implies that there are extremely efficient and highly parallelizable algorithms for optimal estimation and likelihood calculation based on noisy measurements $y_s \in \mathcal{R}^p$ of the process of the form:

$$y_s \;=\; C_s x_s + v_s \tag{2}$$

where $v_s \sim \mathcal{N}(0, R_s)$ and the matrix C_s can specify, in a very general way, measurements taken at different times or spatial locations *and* at different scales [2, 15, 13, 14, 34]. For example, as mentioned in the Introduction, the extension of one of the optimal estimation algorithms to 2-D and quadtrees is applied in [35] to develop a new *scale-recursive* approach to dense motion-field estimation in image sequences that is considerably faster than previously developed algorithms. In addition, the likelihood calculation algorithm can be used, together with the results presented here, for texture identification [34, 36]. An important point about these algorithms, which is of particular significance for 2-D processing, is that they are *recursive* and *not* iterative, and in fact have *constant* complexity per data point or pixel. This is in sharp contrast to the usual iterative algorithms associated with the processing of MRF's [25].

2.2 General Multiscale Models

As we indicated in the preceding section, a basic property of the model (1) is the Markovianity of the state with respect to the ordering structure defined by the dyadic tree. More precisely, let $\Upsilon_s^i, i = 1, 2, 3$ denote the three subsets of states which correspond to viewing node s as a boundary between the three subsets of nodes corresponding to paths leading towards the parent and two offspring nodes[7]. Then,

$$p_{\upsilon_s^1,\upsilon_s^2,\upsilon_s^3|x_s}(\Upsilon_s^1, \Upsilon_s^2, \Upsilon_s^3|X_s) \;=\; p_{\upsilon_s^1|x_s}(\Upsilon_s^1|X_s)p_{\upsilon_s^2|x_s}(\Upsilon_s^2|X_s)p_{\upsilon_s^3|x_s}(\Upsilon_s^3|X_s) \tag{3}$$

By requiring only this property to hold, we obtain a much wider class of processes than that given by (1), but still retain the essential properties leading to the efficient algorithms

[7] We stress the difference here between subsets of nodes (e.g. $\{s_1, s_2, \cdots\}$) subsets of states (e.g. $\{x_{s_1}, x_{s_2}, \cdots\}$).

mentioned above. In particular, the property (3) not only implies that the tree processes are Markov in scale, from coarse-to-fine, but also that the conditional pdf of the state at node s, given the states at all previous scales, depends *only on the state at the parent node* $s\bar{\gamma}$:

$$p_{x_s|x_\sigma,m(\sigma)<m(s)}(X_s|X_\sigma, m(\sigma) < m(s)) \quad = \quad p_{x_s|x_{s\bar{\gamma}}}(X_s|X_{s\bar{\gamma}}) \tag{4}$$

Such tree processes are naturally defined by specifying the parent-offspring conditional pdf's, along with a pdf for the state at the root node of the tree. A simple example of a stochastic process in this general class is the following discrete-state stochastic process $x_s \in \{0, 1, \cdots, L\}$ with parent-offspring conditional probability mass functions given by:

$$p_{x_s|x_{s\bar{\gamma}}}(X_s|X_{s\bar{\gamma}}) \quad = \quad \begin{cases} \theta_{m(s)} & \text{if } X_s = X_{s\bar{\gamma}} \\ (1 - \theta_{m(s)})/L & \text{if } X_s \neq X_{s\bar{\gamma}} \end{cases} \tag{5}$$

where $p_{x_0}(X_0) = 1/(L + 1)$ for $X_0 \in \{0, 1, \cdots, L\}$ and $\theta_{m(s)}$ is a number between 0 and 1 which may vary with scale $m(s)$. A class of processes with this structure and defined on a quadtree has been proposed by Bouman for segmentation applications [8, 9].

Finally, we stress that while (3) implies that a tree process is Markov in scale, the set of states x^m at scale m, viewed as a sequence of length 2^{m-1} is *not* Markov for an arbitrarily chosen set of parent-offspring pdf's. This point can be appreciated by, for example, computing the joint pdf for the four values at the third level of the multiscale process given by (5), and directly checking the conditions required for Markovianity of the single level sequence[8]. However, as we show in the next section, the parent-offspring conditional pdf's *can* be chosen such that the finest level of the tree process can be used to represent *any* 1-D Markov or reciprocal process, with higher levels in the tree corresponding to representations of the process at coarser resolutions.

3 Representation of 1-D Reciprocal and Markov Processes

In this section we describe the basic properties of reciprocal processes in one dimension, introduce and develop representations of reciprocal processes in terms of multiscale stochastic models, and present several examples.

3.1 1-D Reciprocal Processes

A reciprocal process is a first-order MRF on the real line. More precisely, a stochastic process $z_t, t \in \mathcal{R}$ is reciprocal[9] if it has the property that the conditional probability distribution of a state in any open interval (T_1, T_2), conditioned on the states outside of this interval, depends only on the boundary states z_{T_1}, z_{T_2} [22, 32]. That is, for $t \in (T_1, T_2)$:

$$p_{z_t|z_\tau,\tau\in(T_1,T_2)^c}(Z_t|Z_\tau, \tau \in (T_1, T_2)^c) \quad = \quad p_{z_t|z_{T_1},z_{T_2}}(Z_t|Z_{T_1}, Z_{T_2}) \tag{6}$$

[8] The process is Markov only if $\theta_{m(s)} = 1/(L + 1)$. In this case, the values of the process at any level in the tree are independent of one another.

[9] The discussion here refers only to *first-order* reciprocal processes. Extension to higher-order processes is straightforward [22].

where $(T_1, T_2)^c$ denotes the complement of the open interval (T_1, T_2). Reciprocal processes defined on the integers $\mathcal{Z}$ satisfy the same property with the continuous interval (T_1, T_2) replaced by the discrete interval $\{T_1 + 1, T_1 + 2, \cdots, T_2 - 1\}$

Reciprocal processes are closely related to the class of Markov processes. A process z_t on $\mathcal{R}$ or $\mathcal{Z}$ is Markov if past and future values of the state are independent given the present. This means that for $t_2 < t_3$:

$$p_{z_{t_3}|z_{t_1}, t_1 \leq t_2}(Z_{t_3}|Z_{t_1}, t_1 \leq t_2) \;=\; p_{z_{t_3}|z_{t_2}}(Z_{t_3}|Z_{t_2}) \tag{7}$$

As discussed in [1, 22], if a process is Markov then it is also reciprocal, whereas reciprocal processes are *not* necessarily Markov.

3.2 Exact Multiscale Representations of 1-D Reciprocal Processes

In the introduction we described a construction of a Brownian motion b_t over the unit interval via mid-point deflection. As we noted, this corresponds precisely to one of the Gaussian multiscale stochastic models described in Section 2. To see this, consider the following process. At the coarsest level, the initial state x_0 is a three-dimensional vector whose pdf is given by the joint pdf for the values of a Brownian motion at the initial, middle and final points of the interval:

$$x_0 \;\equiv\; \begin{bmatrix} b_0 \\ b_{0.5} \\ b_1 \end{bmatrix} \sim \mathcal{N}(0, P_0) \tag{8}$$

$$P_0 \;=\; \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix} \tag{9}$$

where we have used the facts that $b_0 = 0$, b_t is an independent increments process, and for $t_1 < t_2$, $b_{t_2} - b_{t_1} \sim \mathcal{N}(0, t_2 - t_1)$.

Choosing a value for x_0 as a sample from this distribution corresponds to the first step in the mid-point deflection construction of Brownian motion. The second step in the mid-point deflection construction is the specification of values for the Brownian motion at the one-fourth and three-fourths points. In the context of our multiscale modeling framework, we define two state vectors at the second level of the dyadic tree in Figure 3, each again a 3-tuple. The state on the left represents the values of the Brownian motion at the initial, one-fourth and middle points of the interval, $[b_0, b_{0.25}, b_{0.5}]$, and the state on the right represents the corresponding values in the right half-interval, $[b_{0.5}, b_{0.75}, b_1]$. The sample at the quarter point is given by linear interpolation of b_0 and $b_{0.5}$, plus a Gaussian random variable with variance equal to the variance of the error in this prediction:

$$b_{0.25} = \frac{1}{2}(b_0 + b_{0.5}) + e_{0.25}, \quad e_{0.25} \sim \mathcal{N}(0, 0.125) \tag{10}$$

Likewise, $b_{0.75}$ is chosen by averaging the end points of the right half-interval, $b_{0.5}$ and b_1, and adding in a random value, independent of, and identically distributed to, the deflection term used to create the sample at the one-fourth point.

The construction of $b_{0.25}$ and $b_{0.75}$ in the multiscale model is precisely the same as the mid-point deflection construction of these values. Values of the process at successively finer sets of dyadic points are generated in the same way. At the m^{th} scale, the values of the process at $t = k/2^m, k = 0, 1, \cdots, 2^m$ are represented with 2^{m-1} state vectors, each containing the values of the process at three points, as shown in Figure 4. At any level, each state is a linear function of its parent, plus an independent noise term. Thus, this construction fits precisely into the multiscale modeling framework given by (1) (see Section 3.3 for the precise formulae for A_s and B_s).

Representation of more general 1-D reciprocal processes via multiscale models is a simple extension of the above idea. To construct a multiscale model for a particular reciprocal process $z_t, t \in [0, 1]$, start by choosing the state at the coarsest level as a sample from the joint distribution $p_{z_0, z_{0.5}, z_1}(Z_0, Z_{0.5}, Z_1)$. This generalizes the choice in the construction above in which the state at the top level is chosen using the Gaussian distribution corresponding to a Brownian motion. The two state vectors at the second level are again the three-dimensional vectors $[z_0, z_{0.25}, z_{0.5}]$ and $[z_{0.5}, z_{0.75}, z_1]$, where values for the half-interval mid-points are chosen as samples from the conditional distributions $p_{z_{0.25}|z_0, z_{0.5}}(Z_{0.25}|Z_0, Z_{0.5})$ and $p_{z_{0.75}|z_{0.5}, z_1}(Z_{0.75}|Z_{0.5}, Z_1)$, respectively. Since the process is reciprocal, $z_{0.25}$ and $z_{0.75}$ are conditionally independent given the state at the first level, and thus the modeling structure fits precisely into the more general non-linear model class described in Section 2.2.

The construction above assumes that the process is defined over a continuous interval. In practice, we are typically concerned with processes z_t on a discrete interval, $t \in \{0, 1, \cdots, T\}$. If $T = 2^N$ for some integer N, then we can use essentially the same construction as for the continuous case above. Specifically, $x_0 \equiv [z_0, z_{T/2}, z_T]$ is a random vector chosen from the appropriate distribution for the process of interest. The states at the second level are $[z_0, z_{T/4}, z_{T/2}]$ and $[z_{T/2}, z_{3T/4}, z_T]$, with the half-interval mid-points again chosen using the appropriate distribution. Since there are only a finite number of points in the discrete process, only a finite number of levels are needed to exactly represent it. In particular, with $T = 2^N$, N levels are required.

There are several observations to be made about the continuous and discrete-time construction we have just described. The first is that there is no fundamental difficulty in choosing a point other than the mid-point at each level in these constructions. For example, in the construction of Brownian motion, starting from the initial set of points represented in the root node state, we could next generate any pair of points on either side of 0.5, e.g. $b_{0.1}$ and $b_{0.7}$. However, the regular structure implied by the choice of mid-points may be of some value for processes such as Brownian motion which have stationary increments, as they lead to models in which the model parameters, such A_s and B_s in (1), have very simple and regular characterizations as a function of node s and scale $m(s)$. This regularity in turn leads to simplifications in the structure of algorithms for estimation and signal processing, requiring fewer distinct gains to be calculated and, if parallel implementation is considered, allowing SIMD (single instruction, multiple data) rather than MIMD (multiple instruction, multiple data) implementations.

Secondly, in discrete-time, there will always be at least some degree of irregularity in the multiscale model if the process is defined over $t \in \{0, 1, \cdots, T\}$ and T is not a power of two. In particular, in such a case the structure of the tree and/or the state needed in the multiscale representation of this process will need to be modified. For example, consider a process defined over $t \in \{0, 1, \cdots, 10\}$. In this case, we can develop a model of the type we have described in which the tree is of non-uniform depth and in which we do not have mid-point deflection at some nodes, as indicated in Figure 5a (e.g. in the generation of the value at $t = 3$ given values at 0 and 5). Alternatively, as shown in Figure 5b, we may be able to achieve some level of (and perhaps complete) symmetry by generating *more* than one new point at some nodes (e.g. in Figure 5b we generate values at both $t = 2$ and $t = 3$ given values at 0 and 5).

Furthermore, as we have indicated previously, while our development has focused on first-order reciprocal processes, the extension to higher-order models is straightforward. Indeed, a K^{th}-order model defined on $t \in \{1, 2, \cdots, K(T+1)\}$, where T is a power of 2, can be accommodated by grouping states at adjacent points into sets of size K. Higher-order models can equivalently be represented by simply redefining the state of the process z_t to be a vector of appropriate dimension.

The representations we have introduced to this point have obvious and substantial levels of redundancy. For example, the value of $z_{T/2}$ appears in the state vector at both nodes at the second level of the multiscale model we have described for discrete-time reciprocal processes. More generally, at the m^{th} level of the model for such a process there are 2^{m-1} state vectors containing a total of $3 \times 2^{m-1}$ values, only $2^m + 1$ of which are distinct. This redundancy is actually of minimal consequence for estimation and likelihood calculation algorithms based on these models. However, it is also easy to eliminate the redundancy using a simple modification of the construction we have described. In particular, we may generate *two* internal points between each pair of points at each stage in the level-to-level recursion, yielding a four-dimensional state vector. For example, if the reciprocal process is defined over $t \in \{1, 2, \cdots, 16\}$, then we can choose the non-redundant set of state vectors illustrated in Figure 6. In this case, a first-order reciprocal process is represented by a process with a *four-dimensional* state. In general, at the m^{th} level of such a representation, there are 2^{m-1} state vectors representing 2^{m+1} distinct values of the process. Again, in the situation where T is not a power of two, some irregularity in the structure will be introduced.

Once we allow ourselves to consider such variants on the original mid-point deflection construction in which more than one new point is generated between each pair of previously constructed points, we see immediately that it is possible to generate multiscale representations on trees that are not dyadic. For example, consider a reciprocal process defined on $t \in \{0, 1, \cdots, 3^N\}$. This process is most conveniently represented on the regular structure of a *third-order* tree, as shown in Figure 7. This flexibility of the modeling framework allows the possibility of considering different tradeoffs in terms of level of parallelization and computational power of individual processors when implementing estimation and likelihood calculation algorithms such as those in [2, 15, 13, 14, 35, 34, 36].

Finally, it is of interest to note that the construction we have described, and its several variants, can be interpreted as a non-iterative Gibbs sampler. The Gibbs sampler introduced in [25] is an iterative algorithm for the generation of sample paths of MRF's on a discrete lattice. For 1-D discrete-time reciprocal processes this procedure reduces to using the nearest neighbor conditional probability functions to construct a Markov chain which has an asymptotic distribution equal to the correct distribution of the process. Specifically, at each step of the procedure we modify the current sample path by replacing the value at some point in time, say t_0 with a random value chosen according to the conditional distribution for the process at that point given the current values of the sample path at $t_0 - 1$ and $t_0 + 1$. By cycling repeatedly through all of the points, the sample path is guaranteed to converge to one with the correct statistics. The procedure is conceptually simple but computationally intensive, since the Markov chain requires many transitions for the probability function to converge. In contrast, in our construction, we successively generate samples at new points (e.g. mid-points) conditioned on values at previously generated points, which are *not* nearest neighbors but rather boundary points that partition the time interval of interest. For this reason, and since we begin at the root node with a decimated set of values with the correct distribution, we are *guaranteed* that at each stage the decimated process that is constructed has exactly the correct distribution. Thus, with this structure we visit each time point only once and construct a sample path non-iteratively.

In fairness, an important point to note here is that if a reciprocal process is specified directly in terms of a Gibbs distribution then the calculation of the nearest neighbor pdf's required in the Gibbs sampler is simple [25]. The question then is whether it is also simple to determine the conditional pdf's — e.g. the pdf for $z_{T/2}$ given z_0 and z_T — needed to implement the non-iterative, multiscale procedures we have described. As we have seen for Brownian motion and as we illustrate further in the examples below, in many cases, including all vector Gauss-Markov processes and L-state Markov chains, closed form expressions can be derived for the multiscale representations. Further 1-D examples corresponding to the Ising model are discussed in [34].

3.3 Examples

In this section we discuss several examples of reciprocal processes and their multiscale representations. The first examples describe multiresolution models for general vector Gauss-Markov processes specified in state-space form and allow us to illustrate the interpretation of these multiresolution models as providing approximations based on non-orthogonal expansions. In particular, our model for Brownian motion corresponds to the use of the so-called "hat" function [42] in this expansion, leading to linear interpolation between dyadic points, while a model for the integral of Brownian motion leads naturally to a multiresolution approximation using *cubic* interpolation.

The second part of this section presents several discrete-state examples, the first of which investigates general L-state Markov chains and allows us to make contact with the models used in [8, 9] for segmentation applications. The second example is a general two-state process, which is used to demonstrate that the class of multiscale models is in fact far *richer* than the class of Markov processes.

3.3.1 Gauss-Markov Processes

Consider a vector Gauss-Markov process defined on the interval $[0, 1]$ and given by[10]:

$$\dot{z}_t = F_t z_t + G_t \mu_t \tag{11}$$

[10] While we focus here on the construction of multiscale models for continuous-time Gauss-Markov processes, an exactly analogous set of calculations can be performed for the discrete-time process $z_{t+1} = F_t z_t + G_t \mu_t$

where $z_0 \sim \mathcal{N}(0, \Pi_0), \mathbf{E}\{\mu_t \mu_\tau^T\} = I\delta_{t-\tau}$ and $\mathbf{E}\{\mu_t z_0^T\} = 0$. Define the state transition matrix as $\Phi(t,\tau)$ and state covariance matrix as $\Pi_t = \mathbf{E}\{z_t z_t^T\}$ [20]. Also, let $\hat{z}_{t_2|t_1,t_3}$ denote the conditional expectation z_{t_2} given the states z_{t_1} and z_{t_3}, and $P_{t_2|t_1,t_3}$ the corresponding covariance. It is easy to show that for $t_1 < t_2 < t_3$:

$$\hat{z}_{t_2|t_1,t_3} = \begin{bmatrix} \Pi_{t_1}\Phi(t_2,t_1)^T \\ \Phi(t_3,t_2)\Pi_{t_2} \end{bmatrix}^T \begin{bmatrix} \Pi_{t_1} & \Pi_{t_1}\Phi(t_3,t_1)^T \\ \Phi(t_3,t_1)\Pi_{t_1} & \Pi_{t_3} \end{bmatrix}^{-1} \begin{bmatrix} z_{t_1} \\ z_{t_3} \end{bmatrix} \tag{12}$$

$$P_{t_2|t_1,t_3} = \Pi_{t_2} - \begin{bmatrix} \Pi_{t_1}\Phi(t_2,t_1)^T \\ \Phi(t_3,t_2)\Pi_{t_2} \end{bmatrix}^T \begin{bmatrix} \Pi_{t_1} & \Pi_{t_1}\Phi(t_3,t_1)^T \\ \Phi(t_3,t_1)\Pi_{t_1} & \Pi_{t_3} \end{bmatrix}^{-1} \begin{bmatrix} \Pi_{t_1}\Phi(t_2,t_1)^T \\ \Phi(t_3,t_2)\Pi_{t_2} \end{bmatrix} \tag{13}$$

Using (12) and (13), we can obtain explicit formulae for the parameters A_s, B_s and P_0 in the multiscale model (1) as follows. Let us identify the abstract index s with a pair of numbers (m, φ) which denote the scale and horizontal shift of the node s, respectively. The horizontal shift φ, running from 0 to $2^{m-1} - 1$, indexes the nodes at scale m. For instance, the root node is associated with the pair (1,0), and the left and right nodes at the first level are associated with (2,0) and (2,1), respectively. With this notation, the state at node s on the tree contains the values of the process z_t at the particular three points:

$$x_s \equiv x_{(m,\varphi)} = \begin{bmatrix} z_{2\varphi/2^m} \\ z_{(2\varphi+1)/2^m} \\ z_{(2\varphi+2)/2^m} \end{bmatrix} \tag{14}$$

From the description of the general construction, the *form* of the matrix A_s in (1) is clear:

$$A_s \equiv A_{(m,\varphi)} = \begin{cases} \begin{bmatrix} I & 0 & 0 \\ K_1 & K_2 & 0 \\ 0 & I & 0 \end{bmatrix} & \text{if } \varphi \text{ is even} \\ \\ \begin{bmatrix} 0 & I & 0 \\ 0 & K_1 & K_2 \\ 0 & 0 & I \end{bmatrix} & \text{if } \varphi \text{ is odd} \end{cases} \tag{15}$$

In particular, if φ is even, then the first and third components of the state x_s in (14) correspond to the *first and second* components of $x_{s\bar{\gamma}}$. Thus, the identity matrices in (15) for φ even simply map the first and second components of $x_{s\bar{\gamma}}$ to the first and third components of x_s. In addition, the mid-point prediction of $z_{(2\varphi+1)/2^m}$ is just a linear function of the first two components of the parent $x_{s\bar{\gamma}}$, which is expressed via the matrices K_1 and K_2 in the second row of (15). The matrix A_s for φ odd is similar, and in fact is just a "shifted" version of A_s for φ even (reflecting the fact that the interpolation down to the state on the right depends on the *last* two components of $x_{s\bar{\gamma}}$).

The gain matrices in (15) can be computed directly from (12). Using standard formulae for the inversion of a block 2×2 matrix, we compute:

$$K_1 = \Phi(t_2,t_1) + \Phi(t_2,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T(\Pi_{t_3} - \Phi(t_3,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T)^{-1}\Phi(t_3,t_1)$$

$$-\Pi_{t_2}\Phi(t_3,t_2)^T(\Pi_{t_3} - \Phi(t_3,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T)^{-1}\Phi(t_3,t_1) \tag{16}$$

$$\begin{aligned} K_2 &= -\Phi(t_2,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T(\Pi_{t_3} - \Phi(t_3,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T)^{-1} \\ &\quad +\Pi_{t_2}\Phi(t_3,t_2)^T(\Pi_{t_3} - \Phi(t_3,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T)^{-1} \end{aligned} \tag{17}$$

where $t_1 = 2\varphi/2^m, t_2 = (2\varphi + 1)/2^m$ and $t_3 = (2\varphi + 2)/2^m$.

Likewise, the matrix B_s in (1) has the following block structure:

$$B_s \equiv B_{(m,\varphi)} = \begin{bmatrix} 0 \\ K_3 \\ 0 \end{bmatrix} \tag{18}$$

where K_3 is any matrix such that $K_3K_3^T = P_{t_2|t_1,t_3}$ and, again, $t_1 = 2\varphi/2^m, t_2 = (2\varphi+1)/2^m$ and $t_3 = (2\varphi + 2)/2^m$. The matrix B_s in (18) reflects the fact that *no* noise is added to the first and third components of the state x_s, (which are simply copied from the preceding level), while noise corresponding to the estimation error (13) is added to the second.

Finally, the initial covariance matrix P_0 associated with the root node state is given by:

$$P_0 \equiv \mathbf{E}\left\{\begin{bmatrix} z_0 \\ z_{0.5} \\ z_1 \end{bmatrix}\begin{bmatrix} z_0 \\ z_{0.5} \\ z_1 \end{bmatrix}^T\right\} \tag{19}$$

$$= \begin{bmatrix} \Pi_0 & \Pi_0\Phi(0.5,0)^T & \Pi_0\Phi(1,0)^T \\ \Phi(0.5,0)\Pi_0 & \Pi_{0.5} & \Pi_{0.5}\Phi(1,0.5)^T \\ \Phi(1,0)\Pi_0 & \Phi(1,0.5)\Pi_{0.5} & \Pi_1 \end{bmatrix} \tag{20}$$

For instance, if z_t is a Brownian motion, then $F_t = 0$, $\Phi(t,\tau) = 1$, $\Pi_t = t$ and:

$$\hat{z}_{t_2|t_1,t_3} = \frac{t_3 - t_2}{t_3 - t_1}z_{t_1} + \frac{t_2 - t_1}{t_3 - t_1}z_{t_2} \tag{21}$$

$$P_{t_2|t_1,t_3} = \frac{(t_2 - t_1)(t_3 - t_2)}{t_3 - t_1} \tag{22}$$

Evaluating these at $t_1 = 2\varphi/2^m, t_2 = (2\varphi + 1)/2^m$ and $t_3 = (2\varphi + 2)/2^m$, or using (16) - (17), we have $K_1 = K_2 = 1/2$. Similarly, from (22) and (18), $K_3 = 1/2^{(m+1)/2}$. The conditional expectation $\hat{z}_{t_2|t_1,t_3}$, which specifies A_s as just described, also provides us with the required formula for interpolating between dyadic sample points at any level in our multiscale representation and hence we can interpret this representation as providing a sequence of multiresolution approximations. For example, Brownian motion provides us with the linear interpolation formula given in (21) and illustrated in Figure 2. This corresponds to a multiresolution linear spline approximation or, as also illustrated in Figure 2, as a non-orthogonal multiresolution decomposition using the so-called "hat" function [42].

As a second example, consider the movement of a particle whose *velocity* is given by a Brownian motion. This motion can be described using the following Gauss-Markov process:

$$\dot{z}_t = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z_t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu_t \tag{23}$$

In (23), the first component of z_t is the particle position and the second component is its velocity. The state transition matrix $\Phi(t,\tau)$ and the state covariance matrix Π_t follow by straightforward calculations. Using these, one can show that the terms $\Pi_{t_1}\Phi(t_2,t_1)^T$ and $\Phi(t_3,t_2)\Pi_{t_2}$ in the leftmost block matrix on the right side of (12) contain only cubic powers of t_2. Note also that the block matrix in the middle of the right side does not depend on t_2. Thus, the interpolation of z_{t_2} between t_1 and t_3 is a cubic polynomial in t_2:

$$\hat{z}_{t_2|t_1,t_3} = \begin{bmatrix} c_1 + c_2t_2 + c_3t_2^2 + c_4t_2^3 \\ c_2 + 2c_3t_2 + 3c_4t_2^2 \end{bmatrix} \tag{24}$$

where from (23), the second component of $\hat{z}_{t_2|t_1,t_3}$ is just the derivative of the first. It is clear from the definition of $\hat{z}_{t_2|t_1,t_3}$ that $\hat{z}_{t_1|t_1,t_3} = z_{t_1}$ and $\hat{z}_{t_3|t_1,t_3} = z_{t_3}$. These two constraints provide four linear equations in the four unknown coefficients in (24), and thus uniquely determine the interpolating function (24). Note that the interpolating polynomial for the first component of the state has a continuous derivative at the knot locations $t = k/2^m, k = 0,1,\cdots,2^m$. The interpolation of the first component of the state is shown in Figure 8 for the first two levels of a sample path of the multiscale realization.

3.3.2 Discrete-State Processes

Next consider a general finite-state Markov process $z_t \in \{1,2,\cdots L\}$ defined over a discrete interval $t \in \{0,1,\cdots,T\}$. The probability structure of the process is completely determined by the initial condition $\Pr[z_0 = k]$ for $k \in \{1,2,\cdots L\}$ and by the one-step transition probabilities $P_{i,j} \equiv \Pr[z_t = i|z_{t-1} = j]$. We define the one-step transition matrix:

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,L} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ P_{L,1} & P_{L,2} & \cdots & P_{L,L} \end{bmatrix} \tag{25}$$

Note that the multistep transition probabilities are given by powers of the matrix[11] P:

$$\Pr[z_{t+\tau} = i|z_t = j] = [P^\tau]_{i,j} \tag{26}$$

Using (26) and Bayes' rule it is straightforward to calculate that for $t_1 < t_2 < t_3$:

$$\Pr[z_{t_2} = j|z_{t_1} = i, z_{t_3} = k] = \frac{[P^{t_3-t_2}]_{k,j}[P^{t_2-t_1}]_{j,i}}{[P^{t_3-t_1}]_{k,i}} \tag{27}$$

These conditional probabilities, in addition to the probability function required for the state at the root node of the tree, namely

$$\Pr[z_0 = i, z_{T/2} = j, z_T = k] = [P^{T/2}]_{k,j}[P^{T/2}]_{j,i}\Pr[z_0 = i] \tag{28}$$

allow us to construct the multiscale representation of the process. Note that (27) is the counterpart of the conditional probability equations for Gauss-Markov processes given in

[11] $[A]_{i,j}$ stands for the (i,j) element of the matrix A.

(12) and (13), and that the pdf for the state at the root node (28) is the counterpart of the initial covariance matrix (20).

One special case of this process is the following:

$$P_{i,i} = \mu \tag{29}$$

$$P_{i,j} = \frac{1-\mu}{L-1}, \quad i \neq j \tag{30}$$

with $\Pr[z_0 = i] = 1/L, i = 1, 2, \cdots, L$. Neighboring states of this process tend to be the same, and when the process does change state, no particular change is preferred. Thus, this model would seem to be a natural one to use in segmentation applications and can in fact be viewed as an alternative to the 1-D multiscale model (5) introduced in [8, 9]. As noted in Section 2.2, (5) does *not* in general produce a Markov chain or reciprocal process at the finest level. On the other hand (29) – (30) *is* a Markov model, with:

$$[P^k]_{i,j} = \begin{cases} (1 + (L-1)\vartheta^k)/L & \text{if } i = j \\ (1 - \vartheta^k)/L & \text{if } i \neq j \end{cases} \tag{31}$$

where $\vartheta = (L\mu - 1)/(L - 1)$.

Using (27), for this example we can write down the transition probabilities for the mid-point deflection model. In particular, assuming that T is a power of two, we can associate the state at node s with the following values of the process:

$$x_s \equiv x_{(m,\varphi)} = \begin{bmatrix} z_{2\varphi T/2^m} \\ z_{(2\varphi T+T)/2^m} \\ z_{(2\varphi T+2T)/2^m} \end{bmatrix} \tag{32}$$

where, as in (14), the pair of numbers (m, φ) denote the scale and horizontal shift of the node s, respectively. Thus, to generate the state at node s, given the state at the parent node $s\bar{\gamma}$, we require the following conditional pdf:

$$\Pr[z_{\frac{2\varphi T+T}{2^m}} = j | z_{\frac{2\varphi T}{2^m}} = i, z_{\frac{2\varphi T+2T}{2^m}} = k] = \begin{cases} \zeta_1\zeta_1/\zeta_2 & \text{if } i = j = k \\ \zeta_1\xi_1/\xi_2 & \text{if } i \neq j = k \\ \zeta_1\xi_1/\xi_2 & \text{if } i = j \neq k \\ \xi_1\xi_1/\zeta_2 & \text{if } i = k \neq j \\ \xi_1\xi_1/\xi_2 & \text{if } i, j, k \text{ distinct} \end{cases} \tag{33}$$

where for $l = 1, 2$, $\zeta_l = (1 + (L-1)\vartheta^{T/2^{m-l+1}})/L$ and $\xi_l = (1 - \vartheta^{T/2^{m-l+1}})/L$.

To gain additional insight concerning the structure of our multiscale models, consider the particular example of a stationary two-state binary process with one-step transition matrix and initial state probabilities equal to:

$$P = \begin{bmatrix} 1-\mu & \eta \\ \mu & 1-\eta \end{bmatrix} \tag{34}$$

For this process one can show that:

$$P^k = \frac{1}{\eta+\mu} \begin{bmatrix} \eta + \mu(1-\eta-\mu)^k & \eta - \eta(1-\eta-\mu)^k \\ \mu - \mu(1-\eta-\mu)^k & \mu + \eta(1-\eta-\mu)^k \end{bmatrix} \tag{35}$$

and thus using (35) one can build multiscale representations for the class of stationary binary Markov processes.

Moreover, the mid-point deflection structure can also be used to generate non-Markov processes on the tree. For instance, consider the following binary mid-point "selection" process defined over $t \in \{0, 1, \cdots 2^N\}$ [43]:

$$\Pr[z_{t_2} = i | z_{t_1} = j, z_{t_3} = k] \quad = \quad \begin{cases} \mu & \text{if } i = j = k \\ 1 - \mu & \text{if } i \neq j \text{ and } j = k \\ 0.5 & \text{if } j \neq k \end{cases} \tag{36}$$

with $\Pr[z_0 = i, z_{2^{N-1}} = j, z_{2^N} = k] = 1/8$ for all i, j, k and where $i, j, k \in \{1, 2\}$ and where t_1, t_2, t_3 comprise any 3-tuple of dyadic points corresponding to one of the state vectors in the multiscale representation. At the coarsest scale of this process, the three components of the state vector x_0 are independent and identically distributed random variables, each equally likely to be 1 or 2. It is easy to show that the process resulting from this construction is *not* Markov in general, and thus we can conclude that the set of binary stochastic processes which can be constructed within the mid-point deflection framework is strictly larger than the class of binary Markov processes over intervals.

In fact, a bit of thought shows that the class of processes realizable by multiscale models is quite a bit larger than the class of Markov chains. Indeed, any binary stochastic process defined over $t \in \{0, 1, \cdots, 2^N\}$ when represented via mid-point deflection has a probability structure which is determined by $4(2^N - 1)$ parameters, corresponding to the required conditional probability functions. In particular, the conditional probabilities $\Pr[z_{t_2} = i | z_{t_1} = j, z_{t_3} = k]$ for specific choices of $t_1 < t_2 < t_3$ are uniquely determined by the four parameters $\lambda_{i,j}, (i, j) \in \{(1, 1), (1, 2), (2, 1), (2, 2)\}$, where:

$$\Pr[z_{t_2} = 1 | z_{t_1} = i, z_{t_3} = j] \quad = \quad \lambda_{i,j} \tag{37}$$

Since the process is represented using an N level tree, there are $2^N - 2$ of these conditional densities which must be specified, corresponding to each of the nodes except the root node. The probability function for the state at the root node requires seven parameters, and thus the total number of parameters to be specified is $4(2^N - 2) + 7$. In contrast, a non-stationary binary Markov process defined over the time interval $t \in \{0, 1, \cdots, 2^N\}$ requires at most $1 + 2 \times 2^N$ parameters (one corresponding to the initial probability, and 2 for each transition from t to $t + 1$, for $t = 0, 1, \cdots 2^N - 1$). Since each of the parameters in each case is a probability, i.e. a number in the interval [0,1], we see that the set of processes arising from N-level multiscale models is in one-to-one correspondence with the $(4(2^N - 2) + 7)$-dimensional unit cube, while the set of non-stationary Markov chains over the same length interval corresponds to the $(1 + 2 \times 2^N)$-dimensional unit cube. Thus, for $N > 1$, Markov processes constitute only a "thin" subset of the entire class of binary processes constructed on the tree.

4 Representation of 2-D Markov Random Fields

In this section we first review a few of the properties of MRF's and then describe how they can be represented exactly using multiscale models. We then use these exact representations to motivate a family of approximate representations for Gaussian MRF's employing 1-D wavelet transforms.

4.1 2-D Markov Random Fields

Markov random fields are a multidimensional generalization of 1-D reciprocal processes [4, 45, 29, 25, 22]. A continuous space stochastic process $z_t, t \in \mathcal{R}^n$ is said to be a Markov random field if the conditional probability distribution of the process at a point in the interior[12] $\Omega \setminus \Gamma$ of a closed set Ω with boundary Γ, conditioned on the values of the process outside of $\Omega \setminus \Gamma$, depends only on the values of the process on the boundary set Γ. That is, for $t \in \Omega \setminus \Gamma$:

$$p_{z_t|z_\tau,\tau\in(\Omega\setminus\Gamma)^c}(Z_t|Z_\tau, \tau \in (\Omega \setminus \Gamma)^c) \quad = \quad p_{z_t|z_\tau,\tau\in\Gamma}(Z_t|Z_\tau, \tau \in \Gamma) \tag{38}$$

The definition for MRF's on discrete lattices requires the specification of the notion of the "boundary" of a set in $\mathcal{Z}^n$ [45, 22]. Typically, this is accomplished through the specification of a *neighborhood system.* The idea is that the probability distribution of z_t, conditioned on the values of the process on the rest of the lattice, depends only on the values of the process in the neighborhood of t:

$$p_{z_t|z_\tau,\tau\in\mathcal{Z}^n\setminus\{t\}}(Z_t|Z_\tau, \tau \in \mathcal{Z}^n \setminus \{t\}) \quad = \quad p_{z_t|z_\tau,\tau\in D_t}(Z_t|Z_\tau, \tau \in D_t) \tag{39}$$

In this paper, we focus on 2-D MRF's, i.e. where $t \in \mathcal{Z}^2$, and in this context there is a hierarchical sequence of neighborhoods frequently used in image processing applications [10]. The *first order* neighborhood of a lattice point consists of its four nearest neighbors (in the Manhattan metric), and the *second-order* neighborhood consists of its eight nearest neighbors. A given neighborhood system implicitly determines the boundary set of any particular region. In particular, given the neighborhood system $D_t, t \in \mathcal{Z}^2$, the boundary Γ of a subset Ω of $\mathcal{Z}^2$ is given by the set of points which are neighbors of elements in Ω, but not elements of Ω.

4.2 Exact Multiscale Representations of 2-D Markov Random Fields

The representations of 1-D reciprocal and Markov processes in Section 3 relied on the conditional independence of regions inside and outside a boundary set, and we use the same idea here to represent MRF's. The multiscale model is identical to that used in the 1-D case, except that it is defined on a *quadtree* instead of a dyadic tree. That is, we consider multiscale models in which s denotes a node on the quadtree depicted in Figure 9 and $\bar{\gamma}$ is a four-to-one operator, i.e. each node is the parent of four descendant nodes at the next level.

Consider now a 2-D MRF z_t defined on a $(2^N + 1) \times (2^N + 1)$ lattice. The construction of reciprocal processes in one-dimension started with the values of the process at the initial, middle and end points of an interval. In two dimensions, the analogous top level description consists of the values of the MRF around the outer boundary of the lattice and along the vertical and horizontal "mid-lines" which divide the lattice into four quadrants of equal size.

[12] The notation $\Omega \setminus \Gamma$ denotes the set of elements in Ω which are not in Γ (in this case, the interior of Ω).

For instance, on a 17×17 lattice, the state vector x_0 at the root node of the quadtree contains the values of the MRF at the shaded boundary and mid-line points shown in Figure 10. The boundary and mid-line points are denoted with $\diamond$ and $\circ$ symbols, respectively. In general, the state at the root node is a $(6 \times 2^N - 3)$-dimensional vector (given some ordering of the boundary and mid-line lattice points). To construct a sample path of the MRF, we begin by choosing a sample from the joint pdf of the MRF values defined on the boundary and mid-line set. This is the 2-D counterpart to choosing a sample from the pdf $p_{z_0, z_{0.5}, z_1}(Z_0, Z_{0.5}, Z_1)$ when constructing a 1-D reciprocal process.

In the 1-D case, transitions from the first to second level consisted of obtaining a sample from the conditional distribution of the state at the mid-points of the left and right half-intervals. In two dimensions, we predict the set of values at the mid-lines in each of the four quadrants. The components of the four state vectors at the second level are illustrated in Figure 11 for the 17×17 MRF. The points corresponding to the state in the north-west corner are shaded, and correspond to a scaled and shifted version of the points at the top level. The boundary points of the north-west state are denoted with open and blackened diamond symbols and the new mid-line points are denoted with open circles. Note that the four states at the second level share the black diamond mid-line points of the state at the first level. This is analogous to the 1-D construction in which the mid-point at the first level corresponds to an end point in *both* states at the second level (cf. Figure 4). Each of the states at the second level consists of points carried down from the root node (namely the diamond boundaries of each of the quadrants in Figure 11 as well as new mid-line points within each quadrant (the open circles in Figure 11). These mid-line values are chosen as samples from their joint conditional distribution, given the state at the root node. The key point here is that given the values of the field around the boundary of each quadrant, the values of the field along the mid-lines of that quadrant are *independent* of the values outside this quadrant. Said another way, the four states at the second level of the tree are conditionally independent given the values of the MRF on their respective boundaries, i.e. given precisely that information captured in the state at the first level. Thus, the values along the new mid-lines at the second level can be chosen independently and in parallel, in analogy to the way the two mid-points in the 1-D representations are chosen.

Now, we can iterate the construction by defining the states at successive levels to be the values of the MRF at boundary and mid-line points of successively smaller subregions. Indeed, by subdividing each quadrant in the same way as we did in going from the first level to the second, at the m^{th} level the 4^{m-1} state vectors each contain the values of the MRF at $6 \times 2^{N-m+1} - 3$ boundary and mid-line points. Note that the dimension of the state varies from level to level, reflecting the obvious fact that the number of points in the boundary of a 2-D region depends on the size of the region. The multiscale representation has N levels, and each of the 4^{N-1} states at level N represent 9 values in a 3×3 square. Because of the Markov property, at each level the states are conditionally independent, given their parent state at the next higher level. Thus, the MRF can be thought of precisely as a multiscale stochastic process, and, in the Gaussian case, this leads to models *exactly* as in (1).

As in the 1-D case, there are several comments to make. First, we have described a construction in which the lattice is square. If the MRF is defined over a non-square lattice, then the same basic approach will work. In particular, all we require is some sequence of subregions whose boundaries eventually become dense in the set of lattice points. Likewise, while our construction applies to first or second order MRF's, higher-order models can be represented by taking as state the values of the process along boundaries and mid-lines of "width" greater than one. Second, just as our 1-D multiscale model has a natural interpretation in terms of decimation — e.g. if the points on the finest scale correspond to integers, i.e. to $\mathcal{Z}$, then at the next most fine scale they correspond to even integers, i.e. $2\mathcal{Z}$ — so does our 2-D model, although it differs from the usual notion of decimation in 2-D. Specifically, if the points on the finest scale correspond to $\mathcal{Z}^2 = \mathcal{Z} \times \mathcal{Z}$, then the usual notion of decimation would be $2\mathcal{Z} \times 2\mathcal{Z}$. In contrast, the notion of decimation associated with our multiscale models yields the set $(2\mathcal{Z} \times \mathcal{Z}) \bigcup (\mathcal{Z} \times 2\mathcal{Z})$ at the next scale.

Indeed, the obvious difference between our multiscale MRF representations and those of [27, 26, 30] is that these latter representations *do* correspond to multiscale representations using the usual notion of decimation. That is, the usual decimation leads to representations of the field at coarser levels which correspond roughly to 2-D lowpass filtered and subsampled versions of that at the finest level. Hence, the interpolating functions which generate a process at the finest level from a coarse scale sample naturally correspond to 2-D Haar scaling functions or more generally to localized interpolation operators such as those commonly used for coarse-to-fine grid transfer in multigrid applications. In contrast, the interpolation functions in our representation naturally correspond, in the case of Gaussian MRF's, to the solutions of specific differential (or partial differential) equations determined by the covariance structure of the process. To see this more clearly, note that the linear spline interpolation formula for Brownian motion given values at two points $z_0 = Z_0$ and $z_T = Z_T$ is simply the solution to the second-order differential equation:

$$\frac{d^2}{dt^2}\hat{z}_{t|0,T} = 0 \tag{40}$$

Similarly the interpolation of the first component of the second-order process (23) corresponds to the solution of:

$$\frac{d^4}{dt^4}\hat{z}_{t|0,T} = 0 \tag{41}$$

given $z_0, \dot{z}_0, z_T$ and $\dot{z}_T$. The 2-D example analogous to the linear spline model for Brownian motion is Laplace's equation $\nabla^2 \hat{z} = 0$ given values of z on the boundary of a square region, while the counterpart to (41), corresponding to a second-order model, would be the solution of a homogeneous biharmonic equation $\nabla^4 \hat{z} = 0$ given boundary values and normal derivatives along the boundary (see, for example [44, 31], for related discussions).

Finally, note that it may not be possible to explicitly calculate the scale-to-scale conditional pdf's required to represent an MRF which is specified in terms of local (in space) conditional pdf's. Indeed, even if this were possible in general, it is unlikely that the exact representations of MRF's would lead to radically more efficient algorithms for signal processing, since in this case the scale-recursive structure comes at the price of a high dimensional state. Nevertheless, these exact MRF representations provide substantial evidence that the multiscale model class is *much* richer than its simple structure suggests. Moreover, as we show in the next section in the context of Gaussian MRF's, they can be used as a guide towards other far more parsimonious multiscale models which not only can be used to represent physical processes of interest, but which also lead to efficient algorithms.

4.3 Approximate Multiscale Representations of 2-D Gaussian Markov Random Fields

In this section we propose a family of approximate representations for Gaussian MRF's that provide low-dimensional alternatives to exact multiscale models. The idea behind the approximate representations is to take as the state not boundaries of regions, but rather some reduced-order representation of them. Conceptually, we would like to retain only those components of the boundary that are required to maintain nearly complete conditional independence of regions. In general, exact conditional independence will be lost unless the entire boundary is kept, but as we discuss and illustrate here and in the next section, in many cases only a small amount of information needs to be retained in order to obtain adequate representations of the important statistical and qualitative features of an MRF.

The basis for our approximation methodology is a change of coordinates in representing the values of MRF's along 1-D boundaries. A family of models can then be generated by making different choices for the set of coordinates to be retained and those to be discarded at each level of the multiscale representation. These models range from being exact (if all coordinates are retained) to increasingly approximate and simple as fewer and fewer coefficients are retained. While one can also imagine using a number of different coordinate transformations, such as 1-D Fourier series or Karhunen-Loève expansions, we have chosen here to make a choice consistent with the self-similar structure of our multiscale representations. That is, we will use the *1-D wavelet transform* to represent the values of our field along 1-D boundaries.

The approximate models are derived from a class of *non-redundant* exact representations for MRF's which are the counterpart of those illustrated in Figure 6 for 1-D Markov and reciprocal processes. In particular, the states at the first and second levels of this exact representation for an MRF defined on a 16×16 lattice are shown in Figures 12 and 13. In a multiscale representation of an MRF defined on a $2^N \times 2^N$ lattice, a state at the m^{th} level represents the values of the MRF at $16(2^{N-m} - 1)$ points. We denote this set of points as Γ_s, and we view it as the union of four mutually exclusive subsets. In particular, consider the 112 points associated with the root node state in Figure 12. We can view these as four sets of 28 points, each of which corresponds to the boundary of one 8×8 quadrant. In general, we can divide Γ_s into four sets of $4(2^{N-m(s)} - 1)$ points in a similar fashion, and we denote these subsets as $\Gamma_{s,i}, i \in \{NW, NE, SE, SW\}$, where the subscripts refer to the spatial location of the subset. With $s = 0$ corresponding to the root node, the four subsets $\Gamma_{0,i}, i \in \{NW, NE, SE, SW\}$ are illustrated in Figure 12 with the symbols:

$$\Gamma_{0,NW} \leftrightarrow \bigtriangledown, \lhd, \rhd, \bigtriangleup, \text{ and combinations of these.} \quad (42)$$

$$\Gamma_{0,NE} \leftrightarrow \diamond \quad (43)$$

$$\Gamma_{0,SE} \leftrightarrow \square \quad (44)$$

$$\Gamma_{0,SW} \leftrightarrow \circ \quad (45)$$

Next, we interpret the set of values $\{z_t, t \in \Gamma_{s,i}\}$ for each of these quadrant boundaries, as four 1-D sequences of length $2^{N-m(s)}$, corresponding to each of the sides of the quadrant boundary. Thus, there are a total of *sixteen* 1-D boundary sequences associated with the set Γ_s, and we denote these as: $\beta_{s,i,j}, i \in \{NW, NE, SE, SW\}, j \in \{hu, hl, vl, vr\}$, where the latter four subscripts refer to the "horizontal, upper", "horizontal, lower", "vertical, left" and "vertical, right", respectively. For instance, for the 16 × 16 lattice, the sequences $\beta_{0,i,j}$ are shown in Figure 12. Note there is overlap in the sequences $\beta_{s,i,j}$. For instance, $\beta_{0,NW,hu}$ and $\beta_{0,NW,vl}$ both contain the value of the process at (0,0), and this fact is reflected in Figure 12 by the presence of both $\bigtriangledown$ and $\triangleright$ at this lattice point.

Let us now consider the simplest of our approximate models. Specifically, we take as the state of the approximate representation just the *averages* of the sequences $\beta_{s,i,j}$. The state at any node then has sixteen components:

$$x_s = \begin{bmatrix} x_{s,NW} \\ x_{s,NE} \\ x_{s,SE} \\ x_{s,SW} \end{bmatrix} \tag{46}$$

where:

$$x_{s,i} = \begin{bmatrix} W_0\beta_{s,i,hu} \\ W_0\beta_{s,i,vr} \\ W_0\beta_{s,i,hl} \\ W_0\beta_{s,i,vl} \end{bmatrix} \tag{47}$$

for $i \in \{NW, NE, SE, SW\}$ and where $W_0\beta_{s,i,j}$ denotes the average of the sequence $\beta_{s,i,j}(k)$. Given the definition of the state (46),(47) (which will be generalized shortly to allow general wavelet transform approximations to the sequence $\beta_{s,i,j}$), the conditional parent-offspring pdf's need to be obtained from the MRF being approximated. Instead of using these directly, we make an additional approximation. Let us define the downshift operators $\alpha_i, i \in \{NW, NE, SE, SW\}$, which are the counterparts of the upshift operator $\bar{\gamma}$ defined previously (see Figure 9). In particular, we denote the four offspring of node s as $s\alpha_i, i \in \{NW, NE, SE, SW\}$, where the subscript refers to the relative spatial location of the offspring. In the *exact, non-redundant representations*, the following relationship holds:

$$\begin{aligned} & p_{z_t,t\in\Gamma_{s\alpha_i}|z_\tau,\tau\in\Gamma_s}(Z_t, t \in \Gamma_{s\alpha_i}|Z_\tau, \tau \in \Gamma_s) = \\ & \qquad p_{z_t,t\in\Gamma_{s\alpha_i}|z_\tau,\tau\in\Gamma_{s,i}}(Z_t, t \in \Gamma_{s\alpha_i}|Z_\tau, \tau \in \Gamma_{s,i}) \end{aligned} \tag{48}$$

for $i \in \{NW, NE, SE, SW\}$. What (48) says is that the conditional pdf for the state at node $s\alpha_i$ depends only on a subset of the values making up the state at the parent node s. For example, in the case of the NW offspring of node s, the state in the exact representation at node $s\alpha_{NW}$ (that is, $z_t, t \in \Gamma_{s\alpha_{NW}}$) depends *only* on the NW component of the state at node s (that is, on the values $z_t, t \in \Gamma_{s,NW}$). Thus, in the exact representation the state at node $s\alpha_{NW}$ is *independent* of the values of the MRF at the points in $\Gamma_{s,NE}, \Gamma_{s,SE}$ and $\Gamma_{s,SW}$, given the values at $\Gamma_{s,NW}$. In contrast, it is *not* true in general in the simple approximate representations just described that the state $x_{s\alpha_{NW}}$ is independent of $x_{s,NE}, x_{s,SE}$ and $x_{s,SW}$, given $x_{s,NW}$. That is, simply knowing the average value of a process along each side of a square region does not completely decorrelate the values of the field inside and outside the region. Nevertheless, in our approximate modeling framework we will make exactly this assumption. More generally and precisely, our approximate modeling methodology yields a sequence of models corresponding to differing resolution approximations to the boundary processes $\beta_{s,i,j}(k)$, where (46) – (47) corresponds to the coarsest of these. Using the same symbols $x_{s,i}$ and x_s to denote the state components and state of any of these models, we construct our model by making the approximation corresponding to assuming that the conditional independence property holds, i.e. that:

$$p_{x_{s\alpha_i}|x_s}(X_{s\alpha_i}|X_s) \quad = \quad p_{x_{s\alpha_i}|x_{s,i}}(X_{s\alpha_i}|X_{s,i}) \tag{49}$$

Since the field being approximated is assumed to be jointly Gaussian, the conditional density function (49) is parameterized by conditional means and covariances as in (12) and (13):

$$p_{x_{s\alpha_i}|x_{s,i}}(X_{s\alpha_i}|X_{s,i}) \quad = \quad \mathcal{N}(X_{s\alpha_i}; \hat{x}_{s\alpha_i}, P_{s\alpha_i}) \tag{50}$$

where:

$$\begin{aligned} \hat{x}_{s\alpha_i} &= \mathbf{E}\{x_{s\alpha_i}|x_{s,i}\} && (51) \\ P_{s\alpha_i} &= \mathbf{E}\{(x_{s\alpha_i} - \hat{x}_{s\alpha_i})(x_{s\alpha_i} - \hat{x}_{s\alpha_i})^T\} && (52) \end{aligned}$$

One can then derive the matrices A_s, B_s and P_0 in the multiscale representation of the random field:

$$\begin{aligned} A_{s\alpha_{NW}} &= [K_{NW}, 0, 0, 0] && (53) \\ A_{s\alpha_{NE}} &= [0, K_{NE}, 0, 0] && (54) \\ A_{s\alpha_{SE}} &= [0, 0, K_{SE}, 0] && (55) \\ A_{s\alpha_{SW}} &= [0, 0, 0, K_{SW}] && (56) \end{aligned}$$

where:

$$K_i \quad = \quad \mathbf{E}\{x_{s\alpha_i} x_{s,i}^T\}(\mathbf{E}\{x_{s,i} x_{s,i}^T\})^{-1} \tag{57}$$

Likewise, $B_{s\alpha_i} B_{s\alpha_i}^T = P_{s\alpha_i}$ and $P_0 = \mathbf{E}\{x_0 x_0^T\}$ The assumption (49) is directly reflected in (53) - (56). In particular, the state $x_{s\alpha_i}$ is a function *only* of the i^{th} component of the parent (cf. (46)). Thus, the assumption in (49) leads to relatively simple level-to-level interpolations. Indeed, if the MRF is *stationary*, from symmetry we see that not only do the parameters A_s, B_s depend only on the *scale* of node s, but also, $K_{NW} = K_{NE} = K_{SE} = K_{SW}$. Thus, in this case, the representations are quite parsimonious, and more importantly, this simple structure, in addition to the substantially reduced dimensionality of the approximate representations, leads to considerable efficiencies for smoothing [13, 14] and likelihood calculation algorithms [34, 36].

As we have indicated, the generalization of the coarsest approximate model, with state given by (46), (47) corresponds to using wavelet transforms to obtain different resolution representations of the sequences $\beta_{s,i,j}(k)$. We utilize the wavelet transform for discrete sequences as described in [6]. The wavelet transform of $\beta_{s,i,j}(k), k \in \{1, 2, \cdots, 2^{N-m(s)}\}$ is a set consisting of a single "scaling" coefficient and $2^{N-m(s)} - 1$ "wavelet" coefficients[13].

[13] To be concrete, we assume that the wavelet transform filter/downsample operations are iterated until the sequence of scaling coefficients, i.e. the downsampled output of the lowpass component of the wavelet filter bank, is of length one. More generally, one could stop at any point in the decomposition.

These are computed recursively according to[14]:

$$f_k^{j-1} = \sum_{n=1}^{n=2M} h_n f_{n+2k-2}^j \tag{58}$$

$$d_k^{j-1} = \sum_{n=1}^{n=2M} g_n f_{n+2k-2}^j \tag{59}$$

where the scaling coefficients and wavelet coefficients are f_k^j and d_k^j respectively, h_n, g_n are impulse responses of quadrature mirror filters [19] of length $2M$, and where $f_k^{N-m(s)+1} = \beta_{s,i,j}(k)$. We say that a p^{th}-order representation of the sequence $\beta_{s,i,j}(k)$ is a set consisting of the scaling coefficient and the wavelet coefficients up to order p in the wavelet expansion, and that a zeroth-order representation is a set consisting of just the scaling coefficient. We denote the operator which maps the sequence $\beta_{s,i,j}(k)$ to its p^{th}-order representation as W_p. Note that if $p = N - m(s)$ the representation is complete, since it contains the scaling coefficient and *all* of the wavelet coefficients. For $p > N - m(s)$ we take $W_p = W_{N-m(s)}$ (i.e. if there are fewer than p scales of wavelet coefficients, we keep all of them).

The generalization of the approximate representation based on averages of the 1-D sequences $\beta_{s,i,j}(k)$ discussed previously now just involves a new definition for the state variables x_s. In particular, we simply replace (47) with:

$$x_{s,i} = \begin{bmatrix} W_p\beta_{s,i,hu} \\ W_p\beta_{s,i,vr} \\ W_p\beta_{s,i,hl} \\ W_p\beta_{s,i,vl} \end{bmatrix} \tag{60}$$

where $W_p\beta_{s,i,j}$ denotes the p^{th}-order representation of the sequence $\beta_{s,i,j}(k)$ (a vector of length 2^p if $p \leq N - m(s)$ and of length $2^{N-m(s)}$ if $p > N - m(s)$). Thus, the state at any given node consists of sixteen components, each a p^{th}-order representation of one of the 1-D boundary sequences $\beta_{s,i,j}(k)$ associated with the state x_s. Using this generalized definition for the state, and making the assumption in (49), the parameters A_s, B_s and P_0 can be again computed in the essentially same way as we did for the simpler approximate models.

Several comments are in order. First, note that a simple generalization of the above representation would be to allow *different* levels of approximation for different components of the boundary sequences (e.g. one might use a p_1^{th}-order approximation for "vertical" boundary sequences $\beta_{s,i,j}, j \in \{vr, vl\}$ and a p_2^{th}-order approximation for "horizontal" boundary sequences $\beta_{s,i,j}, j \in \{hu, hl\}$). Examples of such a generalization will be given in the next section in the context of approximate representations for MRF texture models.

Second, note that even if all of the wavelet coefficients are retained at all levels (i.e. if the boundary representations are complete), the representation we have just described will be exact only if the GMRF is Markov with respect to either a first or second-order neighborhood. As we have discussed, higher-order neighborhoods lead to thicker boundaries, and this leads naturally to the idea of taking wavelet expansions of boundaries of width

[14] Our notation is slightly different from that in [6]. In particular, in [6], increasing superscript j corresponds to *lower* levels in the decomposition (i.e., *fewer* wavelet and scaling coefficients), while here it corresponds to *higher* levels.

two or more, and utilizing these as the state. With this expanded family, the approximate representations can be made exact for any GMRF by keeping complete wavelet expansions of all boundary sequences $\beta_{s,i,j}(k)$ at all scales.

Third, not only has dimensionality been reduced in going from the exact to the approximate representations, but it has, in fact, been made *constant* at the first $N-p$ levels of the quadtree, where p is the order of the approximation and the MRF is defined on a $2^N \times 2^N$ lattice. In particular, the dimension of the state at node s is equal to 16×2^p, for $m(s) \leq N-p$. When $m = N-p$, the boundary sequence representations are *complete* and the dimension of the state drops by a factor of 2 at each level thereafter.

Finally, the order of the approximations required to achieve a desired level of fidelity in the approximate model depends, of course, on the statistical structure of the specific GMRF under study. In the next section we present examples which illustrate this for a particular GMRF and a number of different approximate representations.

5 Examples of Approximate 2-D Gaussian MRF Representations

In this section we present examples of multiscale approximate representations of a particular Gaussian MRF. GMRF's have been widely used in the context of texture representation [10, 12, 17, 16, 37] and correspond to the following 2-D autoregressive model [11, 29]:

$$z_{i,j} = \sum_{(k,l)\in D} h_{k,l} z_{i-k,j-l} + e_{i,j} \tag{61}$$

where $h_{k,l} = h_{-k,-l}$, D is the neighborhood [22] around the origin $(0,0)$, the Gaussian driving noise $e_{i,j}$ is a locally correlated sequence of random variables, and $(i,j) \in \{0,1,\cdots,T_1-1\} \times \{0,1,\cdots,T_2-1\}$. In addition, as in [11, 10], we interpret the lattice as a *toroid*, i.e. the independent variables (i,j) in (61) are interpreted modulo (T_1,T_2). For instance, the first-order neighborhood of lattice site $(0,0)$ is given by the set $\{(1,0),(0,1),(0,T_2-1),(T_1-1,0)\}$. Finally, the correlation structure of the driving noise is given by:

$$\mathbf{E}\{e_{i,j}e_{i-k,j-l}\} = \begin{cases} \sigma^2 & \text{if } k=l=0 \\ -\sigma^2 h_{k,l} & \text{if } (k,l)\in D \\ 0 & \text{if } (k,l)\notin D \end{cases} \tag{62}$$

and has the property that $\mathbf{E}\{e_{i,j}z_{k,l}\} = \sigma^2\delta_{i,k}\delta_{j,l}$. Using this latter property, along with the fact that the random field is Gaussian, one can prove that the autoregressive model above *does* imply that $z_{i,j}$ is an MRF [45]. We refer to (61) as an n^{th}-order MRF model if the set D corresponds to the n^{th}-order neighborhood.

The specific statistics and correlations (as in (53) – (56)) required to construct our multiscale models can be computed efficiently using 2-D FFT's because of the fact that correlation matrices for these random fields, assuming lexicographic ordering, are block circulant with circulant blocks and hence these random fields are *whitened* by the 2-D Fourier transform [29]. Indeed, as described in [34], the structure of the approximate representations and the stationarity of the GMRF allow us to compute the required correlations with only 2^p 1-D Fourier transform operations per level of the representation, where p is the order of the approximation. Furthermore, these calculations need only be performed *once*, since they are used simply to determine the parameters in the multiscale approximate model.

Figure 14a illustrates a sample path of a fourth order GMRF corresponding to a "wood" texture [12], and three approximations of this MRF based on the Haar wavelet are shown in Figures 14b - 14d. This texture clearly has a very asymmetric correlation structure, and thus we represent the vertical and horizontal boundary with different levels of approximation. In Figure 14b, the horizontal and vertical boundaries are represented with second and zeroth-order approximations respectively. The boundary effects apparent in Figure 14b are a direct result of the fact that (48) does not hold for the approximate representations, i.e. values of the MRF in distinct quadrants are *not* independent given incomplete information about the boundary and mid-line values. In Figures 14c and 14d, the horizontal boundaries are represented with fourth and sixth-order approximations, respectively, whereas the vertical boundary is again represented with a zeroth-order approximation. As the complexity of the representation increases, the sample paths of the approximate random fields have fewer boundary effects. The approximate representations used to generate Figures 14c and 14d appear to accurately represent the qualitative and statistical features of the MRF. An interesting point here is that the level of representation only needs to be increased in one direction to obtain an excellent representation of the field. Also, the neighborhood of this MRF is fourth-order and thus double width boundaries would be needed in an exact representation. The fields shown in Figures 14b to 14d, however, use only the thinner boundaries in forming states. Several experiments were done in which we used the double width boundaries in forming states for models analogous to those in Figures 14b to 14d. It was found, however, that there were no visual differences between the single and double width approximate representations. Likewise, approximations of the "wood" texture based on the Daubechies 8 wavelet [19] were also visually identical to their Haar-representation counterparts. That is, at least for this example, and for the others we have examined, the critical issue in model fidelity appears to be model order rather than the particular choice of the wavelet used. Furthermore, as these examples indicate, we can achieve quite high quality results with low-order models, which in turn lead to extremely efficient algorithms as in [15, 13, 14, 35, 34, 36].

6 Discussion and Conclusions

In this paper, we have shown how to represent reciprocal and Markov processes in one dimension and Markov random fields in two dimensions with a class of multiscale stochastic models. This modeling structure provides a framework for the development of efficient, scale-recursive algorithms for a variety of statistical signal processing problems. The representations in 1-D rely on a generalization of the mid-point deflection construction of Brownian motion. In 2-D, we introduced a "mid-line" construction which leads to a class of models with scale-varying dimension. In addition, we also introduced a class of multiscale *approximate* MRF representations based on 1-D wavelet transforms. This family allows one to tradeoff complexity and accuracy of the models, and provides a framework for the development of extremely efficient estimation and likelihood calculation algorithms. An example demonstrated that for relatively low-order models, an approximate model which retains most of the qualitative and statistical features of the original MRF can be obtained.

We feel that the results presented in the preceding section, together with the substantial flexibility of the multiscale modeling framework, demonstrate the promise of this framework for image and multidimensional signal processing. Indeed practical applications of this framework are already emerging, as in the segmentation and image sequence processing applications described in [8, 9, 35]. In addition, in [34, 36] we demonstrate the superior performance of likelihood-based texture identification methods using low-order versions of the models introduced in Section 4 — where by "superior" we mean that the algorithm based on our multiscale models has significantly better probability of error characteristics than well-known methods such as those in [10], and achieves virtually the same performance as the truly optimal GMRF-based likelihood ratio test, which, except in special cases, is prohibitively complex computationally in problems of even moderate size.

References

[1] K. Abend, T. Harley, and L. Kanal. "Classification of binary random patterns". *IEEE Transactions on Information Theory*, 11:538–544, 1965.

[2] M. Basseville, A. Benveniste, K. Chou, S. Golden, R. Nikoukhah, and A. Willsky. "Modeling and estimation of multiresolution stochastic processes". *IEEE Transactions on Information Theory*, 38:766–784, 1992.

[3] R. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1982.

[4] J. Besag. "Spatial interaction and the statistical analysis of lattice systems". *Journal of the Royal Statistical Society B*, 36:192–225, 1974.

[5] J. Besag. "On the statistical analysis of dirty pictures". *Journal of the Royal Statistical Society B*, 48:259–302, 1986.

[6] G. Beylkin, R. Coifman, and V. Rokhlin. "Fast wavelet transforms and numerical algorithms I". *Communications on Pure and Applied Mathematics*, 44:141–183, 1991.

[7] A. T. Bharucha-Reid. *Elements of the Theory of Markov Processes and Their Applications.* McGraw-Hill, 1960.

[8] C. Bouman and M. Shapiro. "Multispectral image segmentation using a multiscale model". In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 565–568, 1992.

[9] C. Bouman and M. Shapiro. "A multiscale random field model for Bayesian image segmentation". *IEEE Transactions on Image Processing*, 1993. To appear.

[10] R. Chellappa and S. Chatterjee. "Classification of textures using Gaussian Markov random fields". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33:959–963, 1985.

[11] R. Chellappa and R. Kashyap. "Digital image restoration using spatial interaction models". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 30:461–472, 1982.

[12] R. Chellappa, B. Manjunath, and T. Simchony. Texture segmentation with neural networks. In *Neural Networks for Signal Processing.* Prentice-Hall, 1992.

[13] K. C. Chou. *A Stochastic Modeling Approach to Multiscale Signal Processing.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, May 1991.

[14] K. C. Chou, A. S. Willsky, and A. Benveniste. "Multiscale recursive estimation, data fusion and regularization". Technical Report LIDS-P-2085, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, 1991. To appear in *IEEE Transactions on Automatic Control.*

[15] K. C. Chou, A. S. Willsky, A. Benveniste, and M. Basseville. "Recursive and iterative estimation algorithms for multiresolution stochastic processes". In *Proceedings of the IEEE Conference on Decision and Control*, 1989.

[16] F. Cohen. "Modeling of ultrasound speckle with application in flaw detection in metals". *IEEE Transactions on Signal Processing*, 40:624–632, 1992.

[17] F. Cohen, Z. Fan, and S. Attali. "Automated inspection of textile fabrics using textural models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:803–808, 1991.

[18] F. Cohen, Z. Fan, and M. Patel. "Classification of rotated and scaled textured images using Gaussian Markov random field models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:192–202, 1991.

[19] I. Daubechies. "Orthonormal bases of compactly supported wavelets". *Communications on Pure and Applied Mathematics*, 91:909–996, 1988.

[20] M. H. A. Davis. *Linear Estimation and Stochastic Control.* John Wiley & Sons, New York, 1977.

[21] H. Derin, H. Elliot, R. Cristi, and D. Geman. "Bayes smoothing algorithms for segmentation of binary images modeled by Markov random fields". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:707–720, 1984.

[22] H. Derin and P. Kelly. "Discrete index Markov type random processes". *Proceedings of the IEEE*, 77:1485–1509, 1989.

[23] P. Doerschuk. "Bayesian signal reconstruction, Markov random fields and X-ray crystallography". Technical Report CICS-P-252, Massachusetts Institute of Technology, Center for Intelligent Control Systems, 1990.

[24] P. Flandrin. "Wavelet analysis and synthesis of fractional Brownian motion". *IEEE Transactions on Information Theory*, 38:910–917, 1992.

[25] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[26] B. Gidas. "A renormalization group approach to image processing problems". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:164–180, 1989.

[27] F. Heitz, P. Perez, and P. Bouthemy. "Parallel visual motion analysis using multiscale Markov random fields". In *Proceedings of the IEEE Workshop on Visual Motion*, Princeton, NJ, October 1991.

[28] F. Jeng and J. Woods. "On the relationship of the Markov mesh to the NSHP Markov chain". *Pattern Recognition Letters*, 5:273–279, 1986.

[29] R. Kashyap and R. Chellappa. "Estimation and choice of neighbors in spatial interaction models of images". *IEEE Transactions on Information Theory*, 29:60–72, 1983.

[30] S. Lakshmanan and H. Derin. "Gaussian Markov random fields at multiple resolutions". In *Markov Random Fields: Theory and Application*. Academic Press, 1993.

[31] B. C. Levy. "Non-causal estimation for Markov random fields". In *Proceedings of the International Symposium MTNS-89, Vol. 1: Realization and Modeling in System Theory*, Basel, 1990. Birkhauser-Verlag.

[32] B. C. Levy, R. Frezza, and A. J. Krener. "Modeling and estimation of discrete time Gaussian reciprocal processes". *IEEE Transactions on Automatic Control*, 35:1013–1023, 1990.

[33] P. Lévy. "Le mouvement Brownien". *Mem. Sc. Math., fasc. 126*, pages 1–81, 1954.

[34] M. Luettgen. *Image Processing with Multiscale Stochastic Models*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, May 1993.

[35] M. Luettgen, W. Karl, and A. S. Willsky. "Efficient multiscale regularization with applications to the computation of optical flow". Technical Report LIDS-P-2115, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, 1992. To appear in *IEEE Transactions on Image Processing*, January 1994.

[36] M. Luettgen and A. S. Willsky. "Likelihood calculations for a class of multiscale stochastic models, with applications to texture discrimination". Technical report, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, 1993. In preparation.

[37] B. S. Manjunath and R. Chellappa. "Unsupervised texture segmentation using Markov random field models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:478–482, 1991.

[38] J. Marroquin. "Optimal Bayesian estimators for image segmentation and surface reconstruction". Technical Report LIDS-P-1456, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, 1985.

[39] L. Onsager. "Crystal statistics 1: A two-dimensional model with an order-disorder transition". *Physical Review*, 65:117 - 149, 1944.

[40] T. Pappas. "An adaptive clustering algorithm for image segmentation". *IEEE Transactions on Signal Processing*, 40:901–914, 1992.

[41] B. Potamianos and J. Goutsias. "Stochastic simulation techniques for partition function approximation of Gibb's random field images". Technical Report JHU/ECE 91-02, Johns Hopkins University Department of Electrical and Computer Engineering, 1991. Also submitted to IEEE Transactions on Information Theory.

[42] G. Strang. "Wavelets and dilation equations: A brief introduction". *SIAM Review*, 31:614–627, 1989.

[43] R. Tenney and A. Willsky. "Multi-resolution estimation for image processing and fusion". In *Proceedings of the AFIT Workshop on Wavelets and Signal Processing, TP-329*, 1992.

[44] A. S. Willsky. "Opportunities and challenges in signal processing and analysis". In *Proceedings of the International Conference on Computer Science and Control*, Paris, 1992.

[45] J. Woods. "Two-dimensional discrete Markovian fields". *IEEE Transactions on Information Theory*, 18:232–240, 1972.

[46] J. Woods and C. Radewan. "Kalman filtering in two dimensions". *IEEE Transactions on Information Theory*, 23:473–482, 1977.

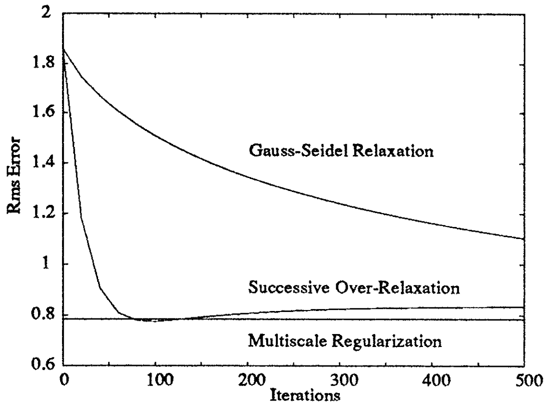

Figure 1: This graph, taken from [35], shows for one example that the multiscale regularization (MR) approach and two smoothness constraint based iterative approaches to the problem of computing optical flow in an image sequence yield comparable rms errors (the rms error corresponding to the non-iterative MR algorithm is shown as a straight line). This result is typical of experiments on several real and synthetic image sequences in [35]. The multiscale approach requires total computation equivalent to 4.2 SOR iterations and hence provides a substantial computational gain.

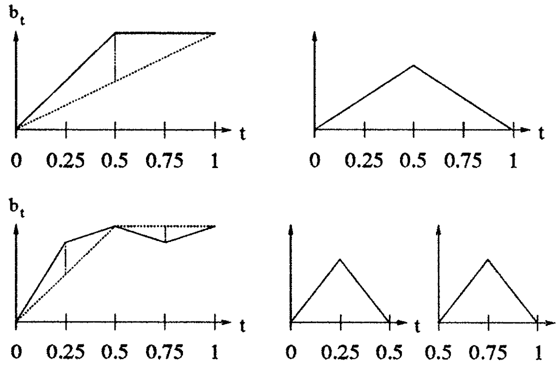

Figure 2: The first two levels of a "mid-point deflection" construction of a Brownian motion sample path are shown on the left. The construction generates a sequence of approximations based on linear interpolations of samples of the Brownian motion at the dyadic points. On the right, the basis functions, integrals of the Haar wavelet, in this construction are shown.

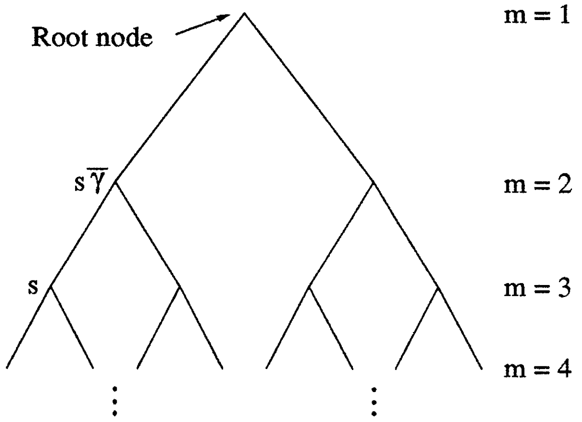

Figure 3: The state vectors in multiscale stochastic models are indexed by the nodes of a *dyadic tree*. The tree is a set of connected nodes, in which each node has two offspring. The parent of node s is denoted $s\bar{\gamma}$ and the scale, or level, of node s is denoted by $m(s)$.

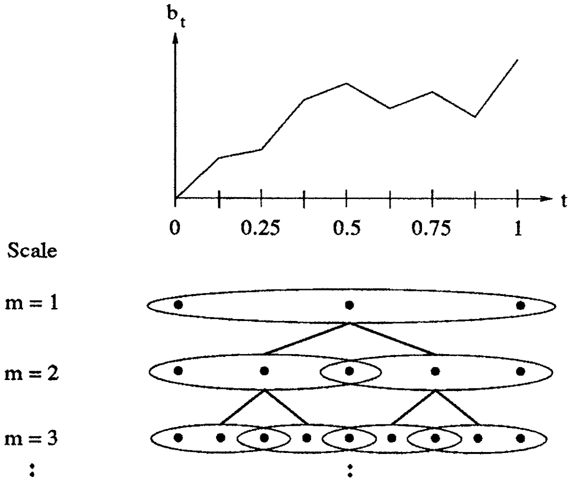

Figure 4: The state vectors for the first three levels of a multiscale model representing Brownian motion, b_t, are illustrated. At the first level, the state is the vector $[b_0, b_{0.5}, b_1]$, which is indicated by the three points at $m = 1$ surrounded by an ellipse. The points are placed directly below the points $t = 0, 0.5$ and $t = 1$ on the graph above to indicate that the state of the multiscale process at the first level consists of the values of the Brownian motion at those three points. Likewise, at lower levels, the states are indicated by sets of three points surrounded by ellipses, with the horizontal location of the points in correspondence with time indices in the graph at the top. At the m^{th} level, there are 2^{m-1} state vectors, each of which consists of the values of b_t at three consecutive dyadic points, and which together represent the values of the Brownian motion at $2^m + 1$ distinct points on the interval $[0, 1]$.

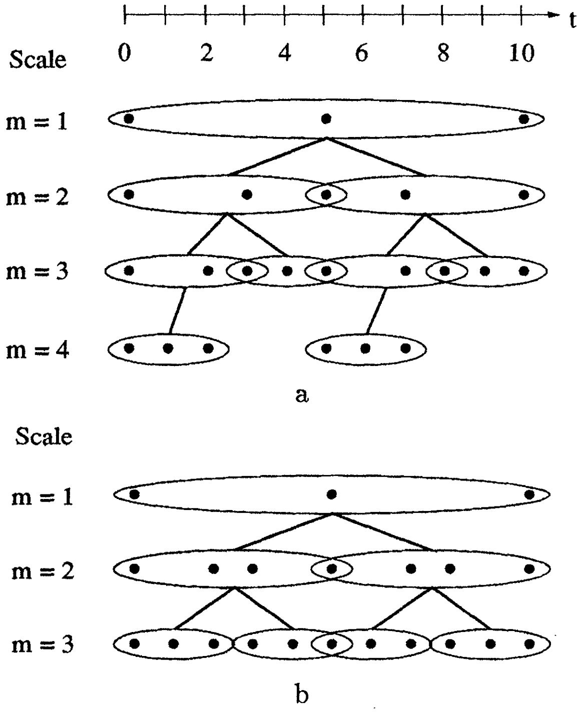

Figure 5: The state vectors are shown for two possible multiscale representations for a reciprocal process defined on a discrete interval of the form $\{0, 1, \cdots, 10\}$. In (a), a dyadic tree with uniform state dimension, but non-uniform depth is used, whereas in (b) a dyadic tree of uniform depth but non-uniform state size is used.

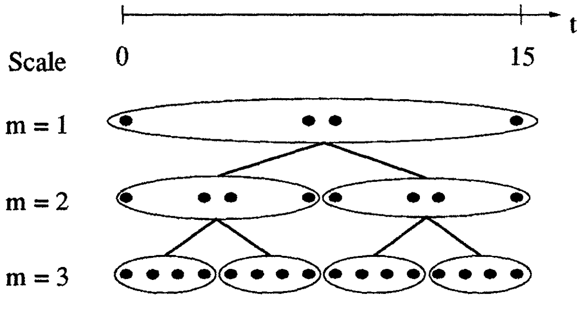

Figure 6: The state vectors are shown for a *non-redundant* multiscale representation of a 1-D reciprocal process. These non-redundant representations, appropriately generalized for the 2-D case, are useful in the context of wavelet-based *approximate* representations of Gaussian MRF's.

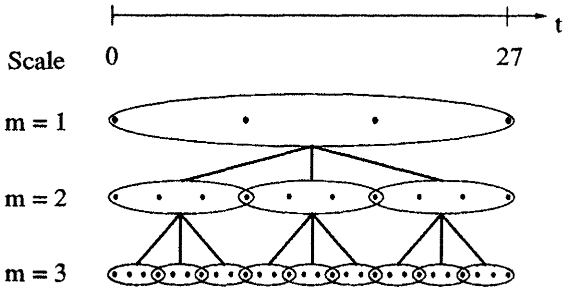

Figure 7: The state vectors are shown for a multiscale representation on a *third-order* tree.

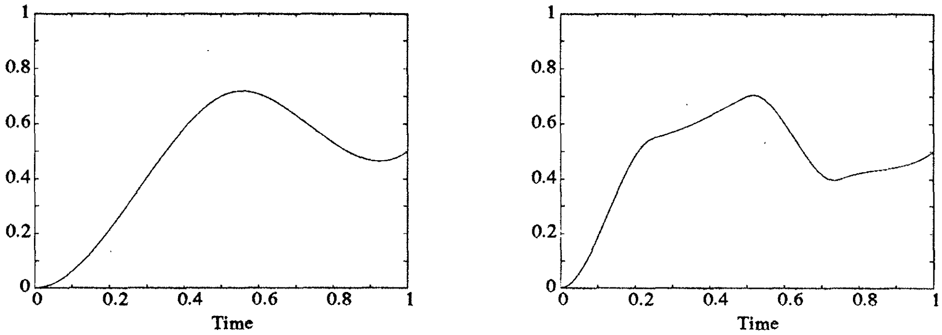

Figure 8: The first two scales in a multiscale representation of a process which is equal to the *second integral* of white noise are shown. The representation consists of samples of the process at dyadic points along with a piecewise-cubic interpolation. Compare these curves with the graphs of Figure 2, which depict the piecewise linear interpolation of the *first* integral of white noise.

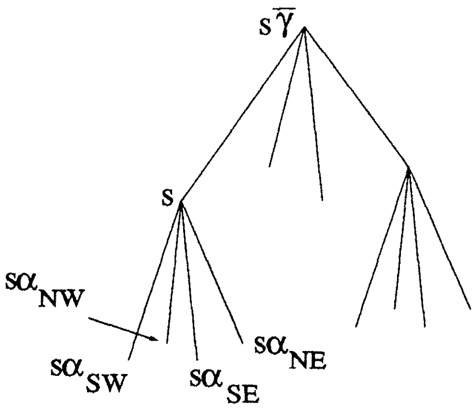

Figure 9: The quadtree structure shown is used for the multiscale representations of Markov random fields (MRF's). Each node of the quadtree has four offspring, denoted $s\alpha_{NW}, s\alpha_{NE}, s\alpha_{SE}, s\alpha_{SW}$. Again, the parent of node s is denoted $s\bar{\gamma}$, and in this case $\bar{\gamma}$ is a *four-to-one* shift operator.

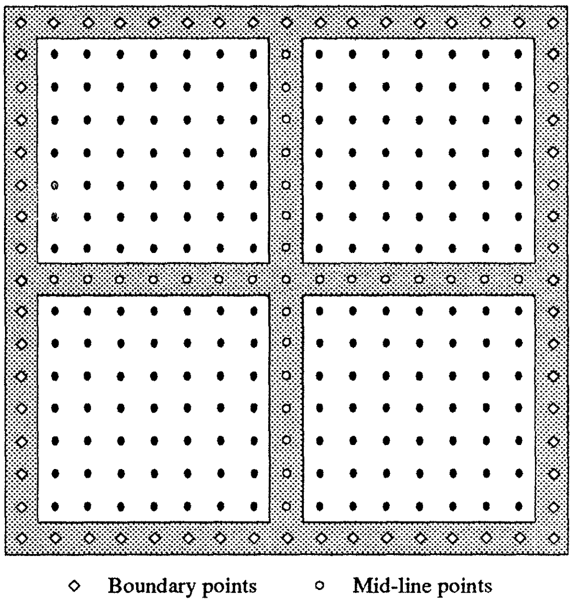

Figure 10: The state vector at the root node in the MRF multiscale representation consists of the MRF values at the boundary and "mid-line" points, shown in the shaded region here for a 17×17 lattice. To construct a sample path of the MRF using the "mid-line" deflection construction, we start by choosing a sample from the joint distribution of the values in the root node state.

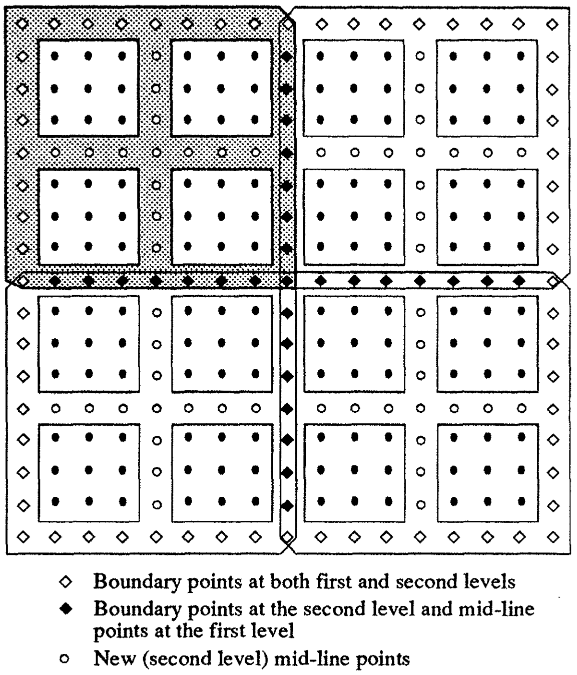
◇ Boundary points at both first and second levels
◆ Boundary points at the second level and mid-line points at the first level
○ New (second level) mid-line points

Figure 11: The components of the four state vectors at the second level of the tree are scaled and shifted versions of the components of the state at the root node. For instance, the state corresponding to the north-west corner at the second level of a representation for an MRF defined on a 17×17 lattice consists of the values of the process at the shaded points. The values of the MRF at the boundary points in these second level states are mapped down from the root node state, and the values at the new mid-lines in each of the four quadrants are chosen independently. In particular, the new mid-line values in any given quadrant are independent of values of the MRF outside that quadrant, given the boundary. Thus, in the construction of a sample path, we can choose values along each of the four sets of new mid-lines independently and in parallel. This process can then be iterated, by defining the states of the multiscale process at lower levels in the quadtree with respect to successively smaller subdomains, and constructing the process (along boundary and mid-line points) independently within each subdomain.

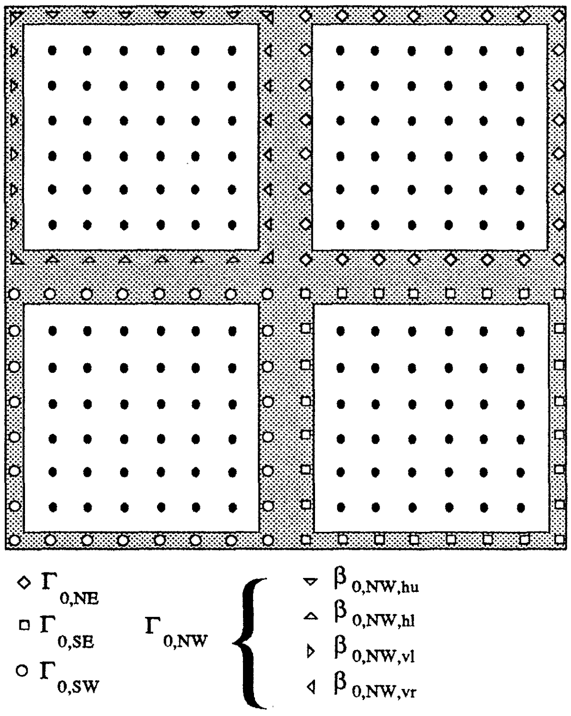

Figure 12: The state at the root node in a *non-redundant* exact multiscale representation of an MRF defined on a 16 × 16 lattice consists of the values of the process at the shaded points. The redundancy in the exact representation is eliminated by generating the values of the process along two mid-lines instead of one. The figure also illustrates the sets $\Gamma_{s,i}$, and the sequences $\beta_{s,i,j}(k)$ defined in the context of approximate representations in Section 4.3. The $\beta_{s,i,j}(k)$ are 1-D sequences corresponding to values of the MRF along boundaries of square subdomains (which, at the first level, are the white areas in the figure). These sequences overlap at the corner points of boundaries. In the figure, this is represented by putting two symbols at the same lattice point, e.g. $\bigtriangledown$ and $\triangleright$ in the upper left corner. The approximate representations take as the state subsets of the coefficients in 1-D wavelet expansions of the $\beta_{s,i,j}(k)$ sequences.

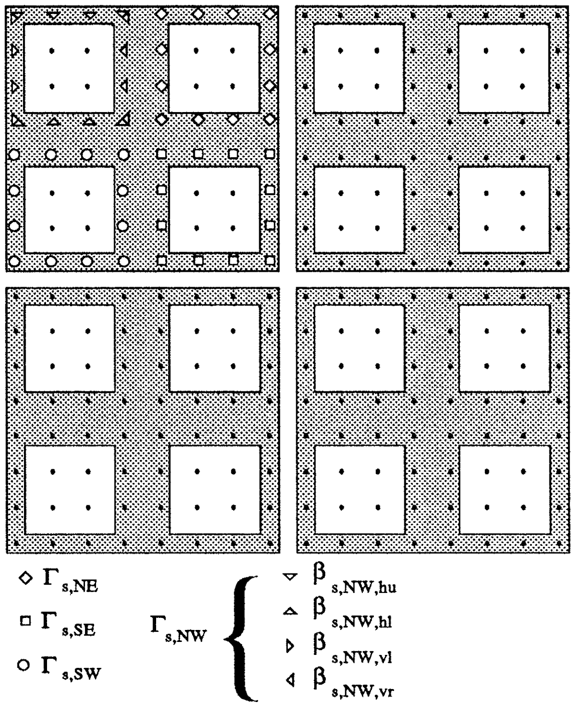

Figure 13: The four states at the second level of the tree in a non-redundant exact multiscale representation are scaled and shifted versions of the state at the root node, and are shown here for an MRF defined on a 16×16 lattice. The state in the north-west corner contains the values of the process at the shaded points in the north-west 8×8 quadrant. With the node s corresponding to this north-west corner state, the sets $\Gamma_{s,i}$ and sequences $\beta_{s,NW,j}$ are illustrated. Note again that the sequences $\beta_{s,i,j}$ overlap.

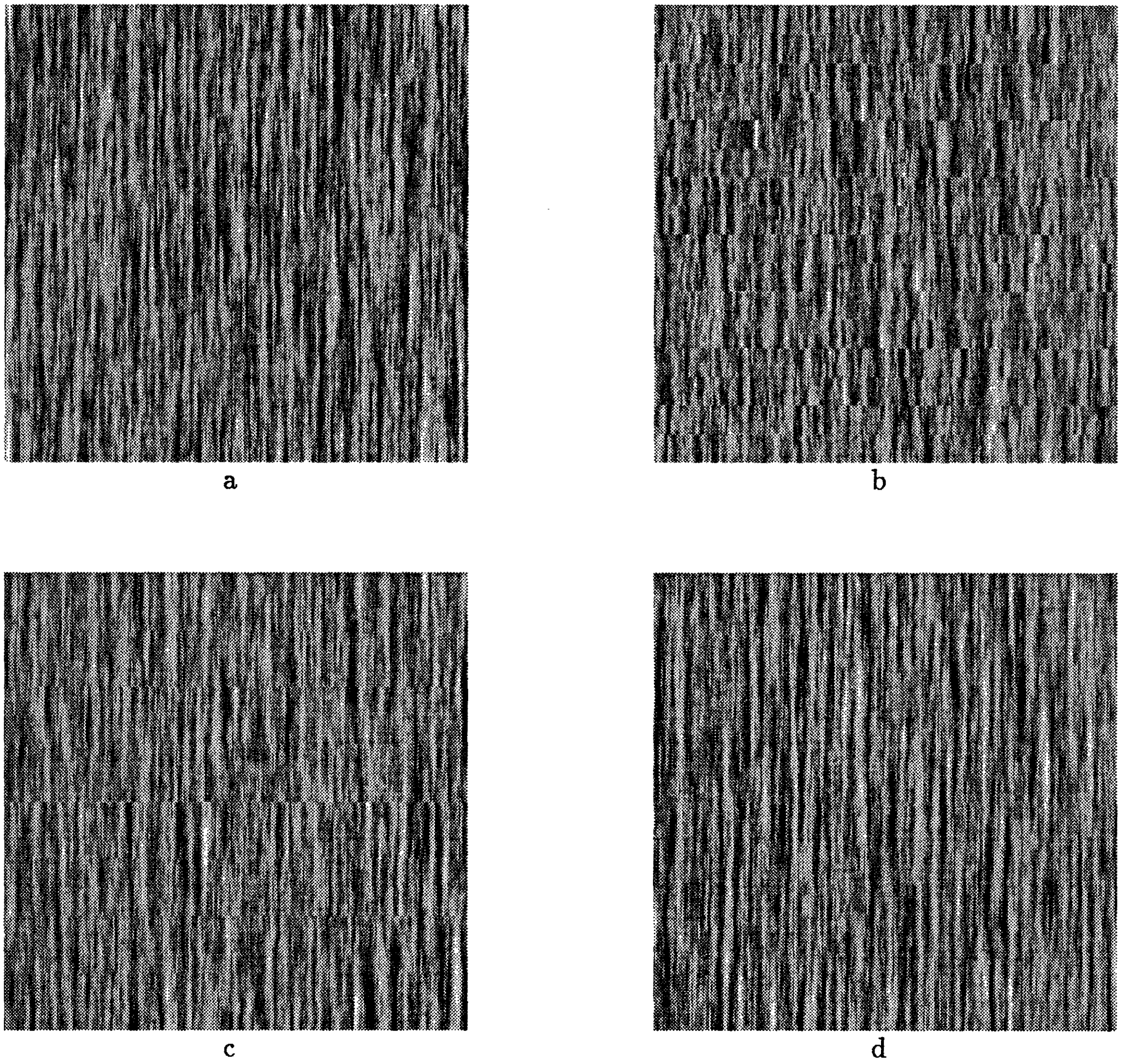

Figure 14: A sample path of a Gaussian MRF representing the "wood" texture of [12] is shown in (a). Parts (b) — (d) illustrate sample paths of approximate representations of the MRF based on the Haar wavelet. The structure of the MRF suggests using approximations which use relatively low order representations of vertical boundaries. The approximate representations used to generate (b) — (d) used zeroth-order representations of the vertical boundaries, and second, fourth and sixth-order representations for the horizontal boundaries, respectively.