September, 1992

LIDS- P 2130

Research Supported By:

Draper Laboratory IR&D agreement
ONR grant N00014-91-J-1004
ARO grant DAAL03-92-G-0115
AFOSR contract 92-J-0002
AFOSR contract F49620-91-C-0047

Multiscale Representations of Markov Random Fields*

Luettgen, M.R.
Karl, W.C.
Willsky, A.S.
Tenney, R.R.

Multiscale Representations of Markov Random Fields[1]

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

September 8, 1992

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. Markov processes in one-dimension and Markov random fields in two-dimensions are widely used classes of models for analysis, design and statistical inference. 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. Our multiscale representations are based on *scale-recursive* models, thus providing a framework for the development of new and efficient algorithms for Markov processes and MRF's. In 1-D, the representation generalizes the mid-point deflection construction of Brownian motion. In 2-D, we use a further generalization to a "mid-line" deflection construction. Our exact representations of 2-D MRF's are of potentially high dimension, and this motivates a class of approximate models based on *one-dimensional* wavelet transforms. We demonstrate the use of these 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. We illustrate how the quality of the representations varies as a function of the underlying MRF and the complexity of the wavelet-based approximate representation.

[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]M.I.T Laboratory for Information and Decision Systems, Cambridge, MA 02139. Electronic mail address: luettgen@athena.mit.edu

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

Contents

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 [10]. 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, 51, 31], and more recently for images. For example, Gaussian fields [64] have been used as image texture models [22, 23, 36, 13, 47, 48], and the more general Gibb's fields have been used as prior models in image segmentation, edge detection and smoothing problems [5, 33, 52, 49]. Causal sub-classes of MRF's, such as Markov Mesh Random Fields [1, 28] and Non-Symmetric Half-Plane Markov chains [35] lead to two-dimensional versions of Kalman filtering algorithms when the fields are Gaussian [65]. In addition, efficient fast Fourier transform algorithms are available for stationary Gaussian fields defined on toroidal lattices [22, 36, 12]. In general, however, Markov random field models lead to computationally intensive algorithms (e.g. stochastic relaxation [33]) for estimation problems. In addition, parameter identification is difficult for MRF models due to the problem of computing the partition function [4, 37, 53]. 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 [16, 17, 18, 19, 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 by Chou et. al. in [16, 17, 18, 19]. Also, Bouman and Shapiro demonstrate how a related multiscale discrete random field [8, 9] leads to an efficient sequential MAP estimator. In this paper, we demonstrate how a simple generalization of the models in [16, 17, 18, 19] 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 not only that it opens the door to the possibility of new and efficient algorithms for MRF models, but also that it suggests that this multiscale modeling framework may be a decidedly superior basis for image and random field modeling and analysis than MRF's both because of the efficient algorithms it admits *and* because of the rich class of phenomena it can be used to describe.

Our multiscale representations of 1-D reciprocal processes and 2-D MRF's rely on a generalization of the mid-point deflection technique for constructing a Brownian motion in one dimension [27, 32, 41]. 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 two boundary points, i.e. the initial and final points, of the interval according to the joint

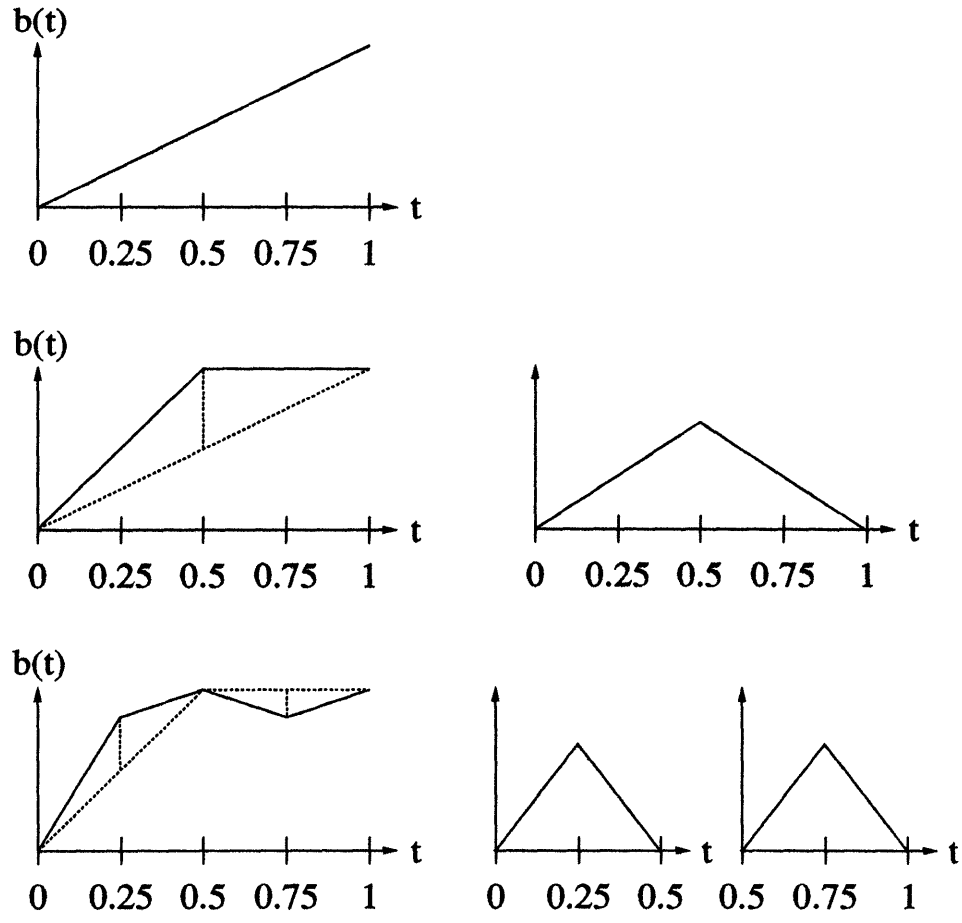


Figure 1: The first three levels of a "mid-point deflection" construction of a Brownian motion sample path are shown on the left. The construction generates a sequence of approximations based on linear interpolations of samples of the Brownian motion at the dyadic points. On the right, the basis functions, integrals of the Haar wavelet, in this construction are shown.

probability distribution implied by the Brownian motion model. We then use these two values to compute the expected value of the process at the mid-point, which is just the average, and then add to that a Gaussian random variable with zero mean and variance equal to the variance of the error in this mid-point prediction. These steps are illustrated in the left side of Figure 1. The process is then continued by using the original boundary points and newly constructed mid-point to predict values of the Brownian motion at the one-fourth and three-fourths points of the interval. 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. The consequence of this observation is that the generation of the one-fourth and three-fourths point values can be carried out *in a parallel and decoupled fashion*: for each we only need the two previously generated points closest to it (e.g. the initial and mid-points for the prediction of the one-fourth point) plus a random value to be added to this prediction that is *independent* of the random value to be added at the other point. Thus, each of these two steps looks *identical* in structure to the original step which generated the mid-point from the two boundary points. 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. At each level in this procedure we generate values at the mid-points of all neighboring pairs of points, in parallel and independently of previously generated points. The structure of all of these steps is identical to that used to generate the original mid-point.

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 1, a sequence of continuous approximations of a Brownian motion is constructed, and the statistics of these approximations converge to those of a Brownian motion [27]. 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 [57] which is the integral of the Haar wavelet [32]. Second, as we will see, the structure of this mid-point deflection construction fits precisely into the multiscale modeling framework developed in [16, 17, 18], and corresponds simply to a particular choice of the parameters in the multiscale model. Moreover, this concept generalizes, allowing us to show that *all* reciprocal and Markov processes in one dimension[4] 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, and most significantly, these same 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 Markov random fields 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.

As a consequence, in addition to the exact representations, we will introduce a family of

[4]Note that this also includes all so-called higher order Markov and reciprocal processes. For example, a second order Markov process, i.e. one for which the value of the process $x(t)$ at time t depends on both $x(t-1)$ and $x(t-2)$, can be represented as a *vector* first order Markov process $[x(t), x(t-1)]^T$, which can then be represented in the manner developed in Section 3.

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.

If wavelet coefficients at all scales are kept along every subdomain boundary for the GMRF, we simply have an exact representation for the GMRF in a transformed basis. However, from wavelet analysis of 1-D stochastic processes, such as in [20, 6, 61, 32], we know that the wavelet transform can achieve high levels of scale-to-scale decorrelation. This result has led, in one dimension, to approximate models and algorithms [20, 66] which neglect the residual correlation, and it suggests here the idea of keeping only a subset of the wavelet coefficients representing each boundary. While this approach would yield a coarse approximation of the GMRF along each boundary, as we move to finer scales, the *length* of each boundary is successively halved, so that "coarse" approximations at fine scales are, in fact, increasingly fine (and, indeed, at fine enough scales represent complete wavelet descriptions). 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. In particular, classes of GMRF's have been widely used to represent natural textures in the context of segmentation and anomaly detection applications [13, 22, 23, 24, 36, 47, 48], and we illustrate how these models can be approximated in our multiscale framework. In addition, we illustrate how the fidelity of the approximation varies with the characteristics of the GMRF being approximated and with the complexity of the approximate representation.

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 MRF's as it possesses advantages both in efficient optimal algorithms *and* in the expressive power of the framework. A number of interesting and substantive problems remain to be investigated — such as the use of alternatives to wavelet expansions such as wavelet packets [62] or lapped orthogonal transforms [45, 46] — and several of these problems, as well as the conclusions of this paper, are presented 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 simple 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 in this section were recently introduced in [16, 17, 18, 19] and describe multiscale Gaussian stochastic processes indexed by nodes on the dyadic tree[5] shown in Figure 2. 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 could interpret the values of the multiscale process as *scaling coefficients* in a Haar wavelet representation of the process at the finest scale. 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 of the process at different levels may correspond to decimated versions of the process at the finest scale. In addition, we may wish to maintain a *set* of values at each node of the tree, i.e. a "state" vector, in order to capture the scale-to-scale memory in a process. For example, one can construct higher order models in which the state includes *both* scaling and wavelet coefficients or *sets* of values of decimated versions of the process. As we will see, our multiscale representations of reciprocal processes and MRF's have a natural interpretation in terms of decimated versions of the processes being modeled, although they can also be interpreted in terms of scaling coefficients corresponding to particular non-orthogonal multiscale representations. Moreover, these models will, in fact, require the use of state vectors at each node on the tree.

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 2. Note that the operator $\bar{\gamma}$ is not one-to-one, but is, in fact, two-to-one since each parent has two offspring. Also, we define the scale of node s, i.e. the level of the node, as $m(s)$. The stochastic tree process $x(s) \in \mathcal{R}^{n_x}$ is then described via the following scale-recursive dynamic model driven by white noise:

$$x(s) = A(s)x(s\bar{\gamma}) + B(s)w(s) \tag{1}$$

[5] As we will see in Section 4 and as is illustrated in Figure 12, the extension to 2-D signals involves, in essence, replacing the dyadic tree by a quadtree.

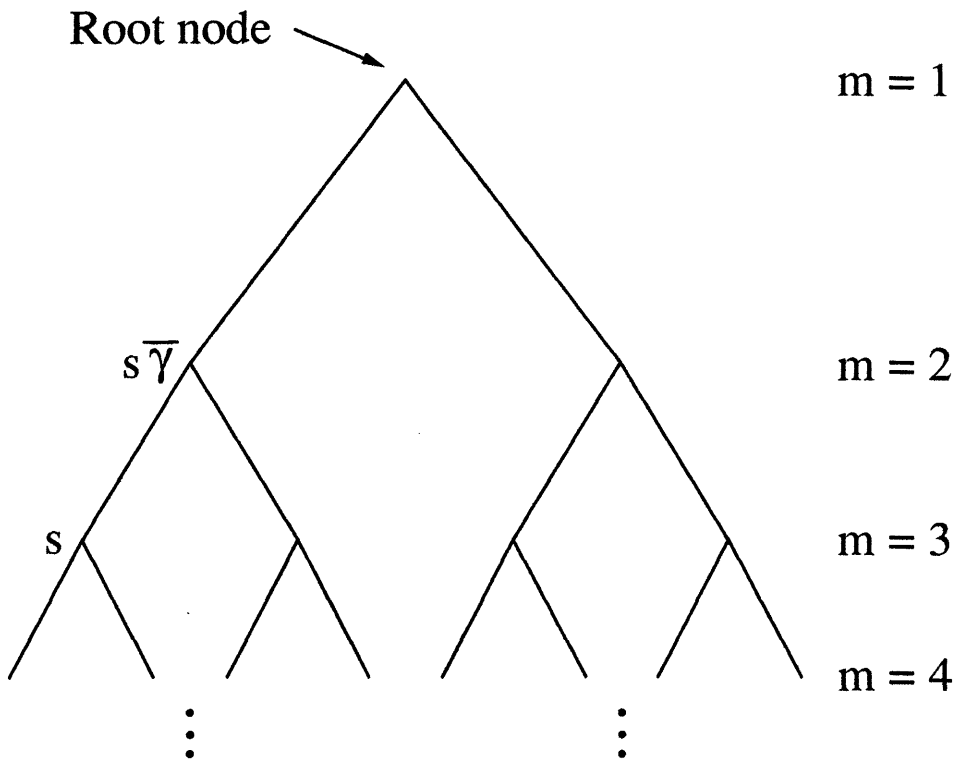


Figure 2: The state vectors in multiscale stochastic models are indexed by the nodes of a *dyadic tree*. The tree is a set of connected nodes, in which each node has two offspring. The parent of node s is denoted $s\bar{\gamma}$ and the scale, or level, of node s is denoted by $m(s)$. Multiscale stochastic processes defined on higher-order trees (see Figures 7 and 12) will also be used.

under the following assumptions[6]:

$$x_0 \sim \mathcal{N}(0, P_0) \tag{2}$$

$$w(s) \sim \mathcal{N}(0, I) \tag{3}$$

where $w(s) \in \mathcal{R}^{n_w}$ and $A(s)$ and $B(s)$ are matrices of appropriate size. The state variable x_0 at the root node of the tree provides an initial condition for the recursion. The driving noise $w(s)$ is white, i.e. $w(s)$ and $w(\sigma)$ are uncorrelated if $s \neq \sigma$, and is uncorrelated with 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. For example,

[6]The notation $x \sim \mathcal{N}(m, P)$ means the random vector x is normally distributed with mean m and variance P. We also sometimes use the notation $p_x(X) = \mathcal{N}(X; m, P)$ to denote the same thing.

if $x(s)$ is scalar and $A(s) = 1$, the values of $x(s)$ at any particular scale have a natural interpretation as providing a piecewise-constant (i.e. Haar) approximation to a signal, and the $B(s)w(s)$ terms then represent additional detail added in progressing from coarse to fine scales [16, 17, 18, 19]. If, in addition, the magnitude of this detail decreases geometrically with scale — i.e. if $B(s) = 2^{-\mu m(s)}$, where μ is a scalar constant — then the process so represented has fractal characteristics [17, 18, 20]. In fact, in general, dependence of $A(s)$ and $B(s)$ on scale $m(s)$ allows one to model self-similar statistical structures or the presence of dominant scales [20], while allowing general dependency on the node s provides additional flexibility including the ability to capture non-stationarities, localized events, etc. 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.

An extremely important property of the scale-recursive model (1) is that not only is it Markov from scale-to-scale but it also is Markov with respect to the partial ordering defined by the dyadic tree structure. That is, conditioned on the value at a parent node, the evolution of the process in the subtree descendant from that node is *independent* of everything outside that subtree. This fact implies that there are extremely efficient and highly parallelizable algorithms for optimal estimation and other signal processing tasks based on this model.

In particular, as developed in [2, 16, 17, 18, 19], there is an extremely efficient algorithm for optimal estimation based on noisy measurements $y(s) \in \mathcal{R}^{n_y}$ of the process of the form:

$$y(s) \;=\; C(s)x(s) + v(s) \tag{4}$$

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. The structure of the algorithm for this problem consists of two scale-recursive sweeps (a fine-to-coarse sweep followed by a coarse-to-fine recursion), each of which follows the structure of the dyadic tree so that there is substantial parallelism and efficiency. For example, the extension of this algorithm to 2-D and quadtrees is applied in [42] 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, a related algorithm, which is also highly parallelizable, can be used to compute the likelihood of the measurements (log of the conditional probability density) given the model parameters [43]. This algorithm can be used, together with the results presented here, for texture identification and segmentation, for example. 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 [33].

2.2 General Multiscale Models

As we indicated in the preceding section, a basic property of the model (1) – (3) is the Markovianity of the state with respect to the ordering structure defined by the dyadic tree. More precisely, two states are conditionally independent, given a state on the path between them:

$$p_{x(s_1),x(s_2)|x(s_3),s_3\in\Gamma_{s_1,s_2}}(X_{s_1}, X_{s_2}|X_{s_3}, s_3 \in \Gamma_{s_1,s_2}) =$$

$$p_{x(s_1)|x(s_3),s_3\in\Gamma_{s_1,s_2}}(X_{s_1}|X_{s_3}, s_3 \in \Gamma_{s_1,s_2})p_{x(s_2)|x(s_3),s_3\in\Gamma_{s_1,s_2}}(X_{s_2}|X_{s_3}, s_3 \in \Gamma_{s_1,s_2}) \quad (5)$$

where $\Gamma_{s,\sigma}$ is defined as the set of nodes in the shortest path between two nodes s and σ (but not including s or σ). By starting with this assumption, and then allowing the probability density function (pdf) for the state at a particular node given its parent and the pdf for the state at the root node to be arbitrary, we obtain a much wider class of processes than that given by (1) – (3). Indeed, we define a Markov tree process as any set of random variables which are indexed by the nodes of a tree and have a joint pdf satisfying (5).

Markov 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 more 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:

$$p_{x(s)|x(s\bar{\gamma})}(X_s|X_{s\bar{\gamma}}) = \begin{cases} \theta_{m(s)} & \text{if } X_s = X_{s\bar{\gamma}} \\ (1-\theta_{m(s)})/L & \text{if } X_s \neq X_{s\bar{\gamma}} \end{cases} \quad (6)$$

$$p_{x_0}(X_0) = 1/(L+1), \qquad \text{for } X_0 \in \{0, 1, \cdots, L\} \quad (7)$$

where $\theta_{m(s)}$ 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 [7, 8, 9]. We will see several other examples of such general Markov tree processes in the next section.

The property (5) implies that the tree processes are Markov in scale, from coarse-to-fine. In particular, we denote the set of states at a given scale, say m_0, as $x^{m_0} \equiv \{x(s)|m(s) = m_0\}$. Then, using (5), the states at a particular scale, given the states at the previous level, are independent of all coarser scales:

$$p_{x^{m_2}|x^{m_1},m_1<m_2}(X^{m_2}|X^{m_1}, m_1 < m_2) = p_{x^{m_2}|x^{m_2-1}}(X^{m_2}|X^{m_2-1}) \quad (8)$$

Indeed, (5) implies 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* $s\bar{\gamma}$:

$$p_{x(s)|x(\sigma),m(\sigma)<m(s)}(X_s|X_\sigma, m(\sigma) < m(s)) = p_{x(s)|x(s\bar{\gamma})}(X_s|X_{s\bar{\gamma}}) \quad (9)$$

In addition, (5) implies that such a process can also be viewed as a Markov random field on the tree. In particular, define the neighborhood set D_s of node s as its three nearest neighbors on the tree[7] (i.e. the parent and two offspring nodes). The general multiscale process described above is then a Markov random field on the tree in the sense that:

$$p_{x(s)|x(\sigma),\sigma\neq s}(X_s|X_\sigma, \sigma \neq s) = p_{x(s)|x(\sigma),\sigma\in D_s}(X_s|X_\sigma, \sigma \in D_s) \quad (10)$$

The most general class of joint probability distribution functions which lead to conditional distributions satisfying (10) is given by the Hammersley-Clifford theorem [4]. Our focus here is on processes which also satisfy the one-sided Markov property (9), because this class of processes leads naturally to efficient scale-recursive algorithms.

[7]Obvious modifications of the neighborhood set must be made for the root node at the top of the tree, which has no parent, and the nodes at the finest level of the tree, which have no offspring.

Finally, we stress that while (5) implies that a tree process is an MRF with respect to the nearest neighbor set on the tree, 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 (6), 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 single level process x^m *is* Markov in the usual sense, and in fact 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 nothing more than a first-order MRF on the real line. More formally, a stochastic process $z(t), t \in \mathcal{R}$ is said to be reciprocal[9] (or bilateral Markov, two-sided Markov or non-causal Markov) if it has the property that the probability distribution of a state in any open interval (T_1, T_2), conditioned on the boundary states $z(T_1), z(T_2)$ is independent of states outside of the interval [29, 38]. That is, for $t \in (T_1, T_2)$:

$$p_{z(t)|z(\tau),\tau\in(T_1,T_2)^c}(Z_t|Z_\tau, \tau \in (T_1,T_2)^c) = p_{z(t)|z(T_1),z(T_2)}(Z_t|Z_{T_1}, Z_{T_2}) \qquad (11)$$

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 $\{T_1 + 1, T_1 + 2, \cdots, T_2 - 1\}$:

$$p_{z(t)|z(\tau),\tau\in\{T_1+1,\cdots,T_2-1\}^c}(Z_t|Z_\tau, \tau \in \{T_1+1,\cdots,T_2-1\}^c) = p_{z(t)|z(T_1),z(T_2)}(Z_t|Z_{T_1}, Z_{T_2}) \qquad (12)$$

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 $t_2 < t_3$:

$$p_{z(t_3)|z(t_1),t_1\le t_2}(Z_{t_3}|Z_{t_1}, t_1 \le t_2) = p_{z(t_3)|z(t_2)}(Z_{t_3}|Z_{t_2}) \qquad (13)$$

As shown in [1], if a process is Markov then it is also reciprocal. On the other hand, reciprocal processes are not necessarily Markov [29], although one can show that essentially all stationary Gaussian reciprocal processes are Markov [40].

[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 [29].

3.2 Exact Multiscale Representations of 1-D Reciprocal Processes

In the introduction we described a construction of a Brownian motion $b(t)$ over an interval, say $[0,1]$, via mid-point deflection [27]. 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:

$$x_0 \equiv \begin{bmatrix} b(0) \\ b(0.5) \\ b(1) \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, P_0) \tag{14}$$

$$P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix} \tag{15}$$

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 the distribution (14) corresponds to the first two steps in the mid-point deflection construction of Brownian motion, the first step being a choice for the two end-point values $b(0), b(1)$, and the second step being construction of the mid-point $b(0.5)$. The next 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 2, 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:

$$b(0.25) = \frac{1}{2}(b(0) + b(0.5)) + e(0.25) \tag{16}$$

$$e(0.25) \sim \mathcal{N}(0, 0.125) \tag{17}$$

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 the deflection term used to create the sample at the one-fourth point:

$$b(0.75) = \frac{1}{2}(b(0.5) + b(1)) + e(0.75) \tag{18}$$

$$e(0.75) \sim \mathcal{N}(0, 0.125) \tag{19}$$

The above construction of $b(0.25)$ and $b(0.75)$ 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

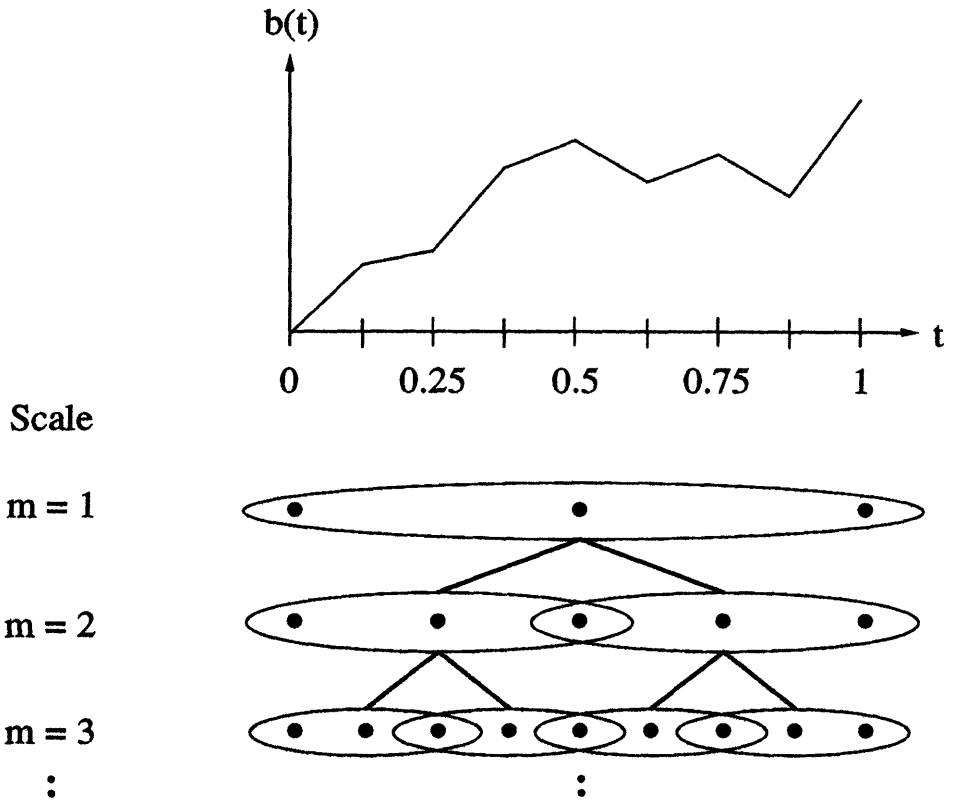


Figure 3: The state vectors for the first three levels of a multiscale model representing Brownian motion, $b(t)$, are illustrated. At the first level, the state is the vector $[b(0), b(0.5), b(1)]$, which is indicated by the three points at $m = 1$ surrounded by an ellipse. The points are placed directly 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 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 $[0, 1]$. The multiscale representation for Brownian motion can be generalized to the class of 1-D reciprocal process, which contains the class of 1-D Markov processes.

values of the process at three points, as shown in Figure 3. 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) – (3) (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:

$$p_{z(0),z(0.5),z(1)}(Z_0, Z_{0.5}, Z_1). \tag{20}$$

This generalizes the choice in (14) 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:

$$p_{z(0.25)|z(0),z(0.5)}(Z_{0.25}|Z_0, Z_{0.5}) \tag{21}$$

$$p_{z(0.75)|z(0.5),z(1)}(Z_{0.75}|Z_{0.5}, Z_1) \tag{22}$$

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 given in (14), 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) – (3) have very simple and regular characterizations as a function of node s and scale $m(s)$ (see, for example, (46),(47)). 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 4a (e.g. in the generation of the value at $t = 3$ given values at 0 and 5). Alternatively, as shown in Figure 4b, 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 4b we generate values at both $t = 2$ and $t = 3$ given values at 0 and 5). Obviously, as in standard discrete signal processing applications in which the FFT is to be used, there are considerable efficiencies to be had if T is a power of 2.

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. The states at different levels of the tree might be as depicted in Figure 5 for a second-order reciprocal process defined over $t \in \{0, 1, \cdots, 18\}$. 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 algorithms based on these models. However, it is also easy to eliminate the redundancy by a simple modification to 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, instead of the process with a three-dimensional state used earlier. 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, a q^{th}-order tree is a set of connected nodes, such that each node has q offspring as in Figure 7. 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 8. 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 algorithms such as those in [2, 16, 17, 18, 19, 42].

Finally, it is of interest to note that the construction we have described, and its several variants, can be interpreted as a non-iterative Gibb's sampler. The Gibb's sampler introduced in [33] is an iterative algorithm for the generation of sample paths of Markov random fields 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 joint distribution of the process. Specifically, at each step of the procedure we modify the current sample path

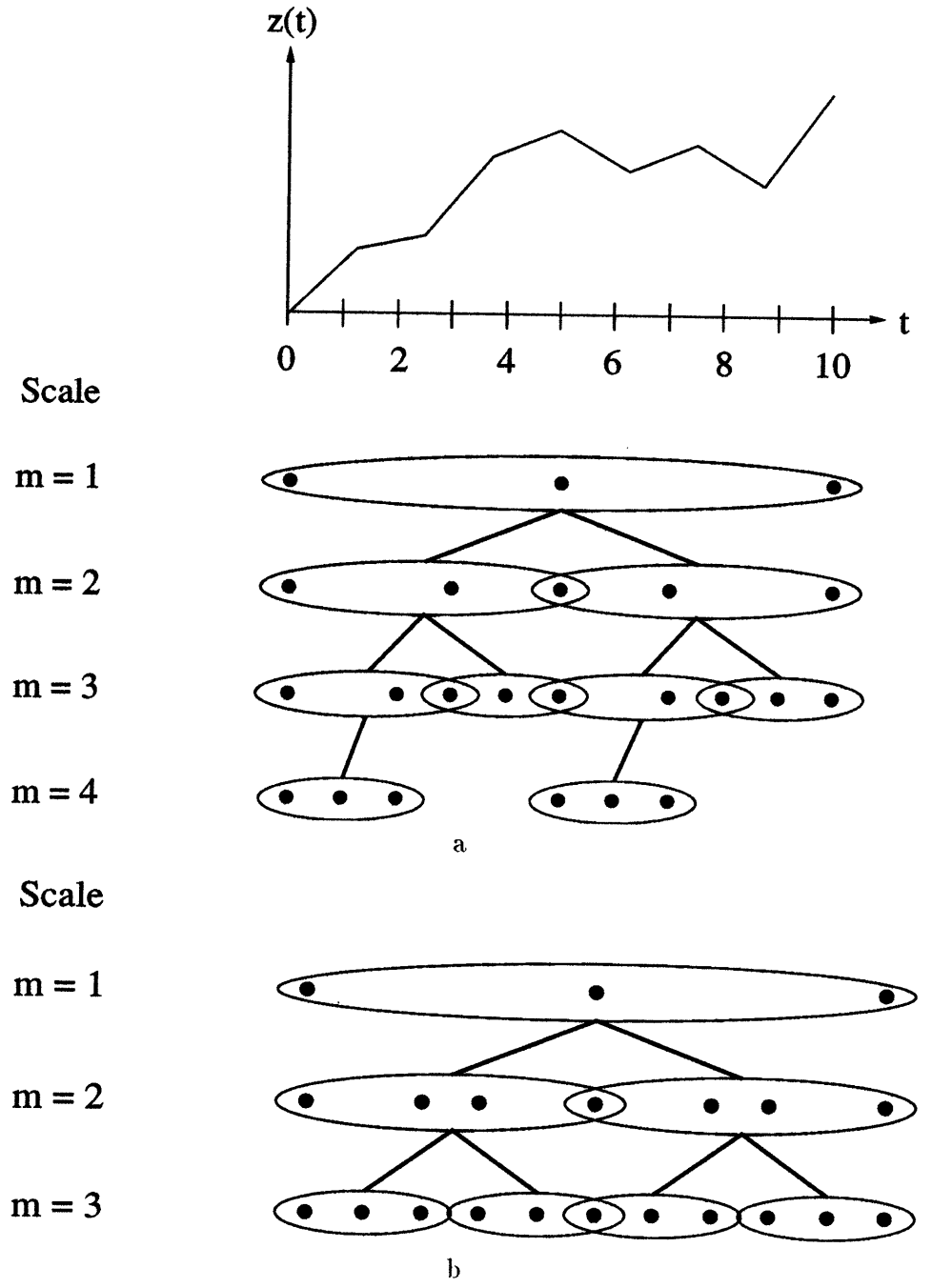


Figure 4: The state vectors are shown for two possible multiscale representations for a reciprocal process which is defined on a discrete interval of the form $\{0, 1, \cdots, 10\}$. In (4a), a dyadic tree with uniform state dimension, but non-uniform depth is used, whereas in (4b) a dyadic tree of uniform depth but non-uniform state size is used.

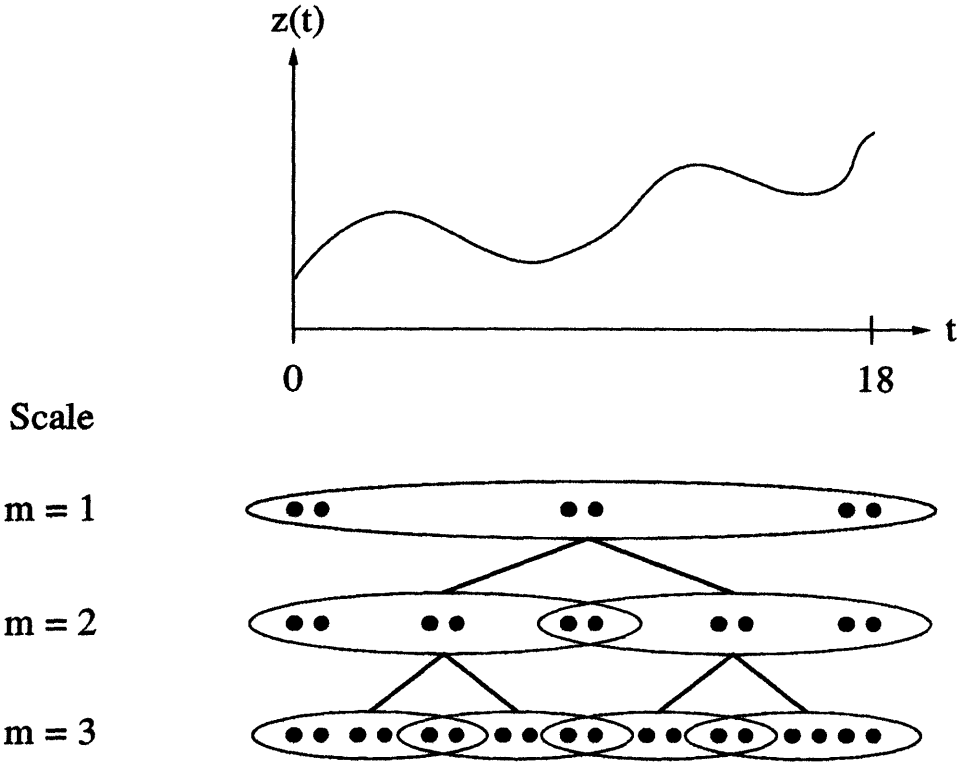


Figure 5: The state vectors are shown for a multiscale representation of a second-order reciprocal process defined on a discrete interval. In this case, the state vectors are composed of three groups of two points, reflecting the fact that the value at the current point, say $z(t_0)$ is independent of the values at all other times, given the *pairs* of nearest neighbors $z(t_0-1), z(t_0-2)$ and $z(t_0+1), z(t_0+2)$.

$z_k(t)$, $t \in \{0, 1, \cdots, T\}$ 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 time points, the sample path is guaranteed to converge to one with the correct joint 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 Gibb's distribution — i.e. in terms of nearest neighbor energy functions (see, for example, [33]) — then the calculation of the nearest neighbor pdf's required in the

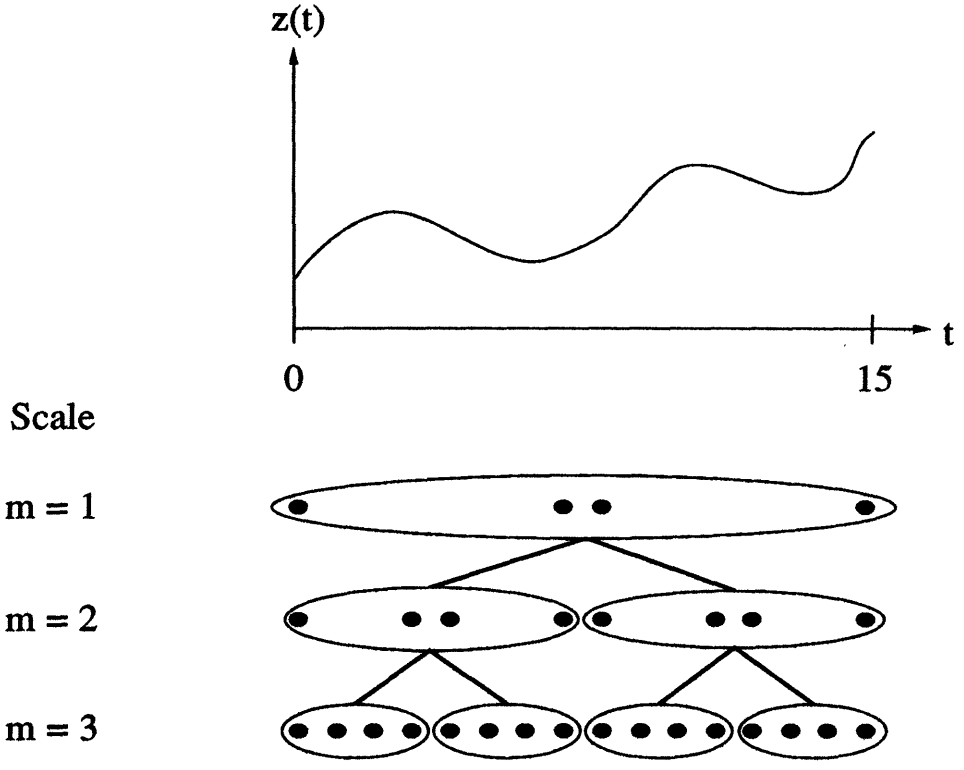


Figure 6: The state vectors are shown for a *non-redundant* multiscale representation of a 1-D reciprocal process. These non-redundant representations, appropriately generalized for the 2-D case, are useful in the context of wavelet-based *approximate* representations of Gaussian MRF's.

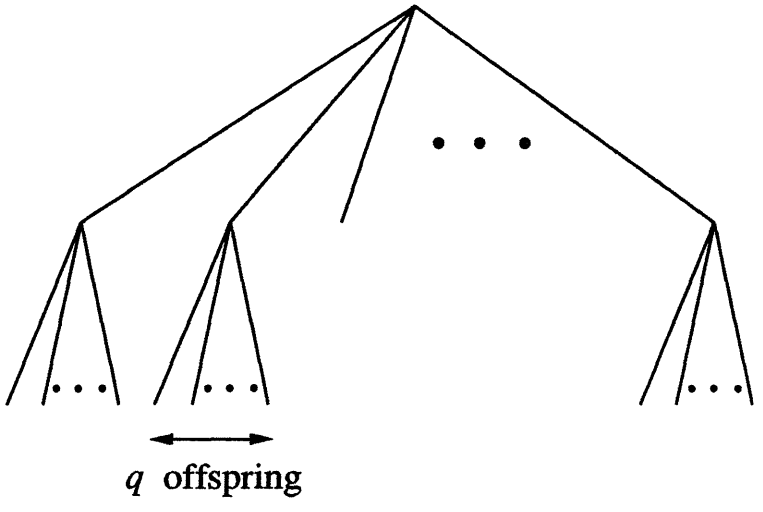


Figure 7: The q^{th}-order tree is illustrated. This is a set of nodes, each of which has q offspring.

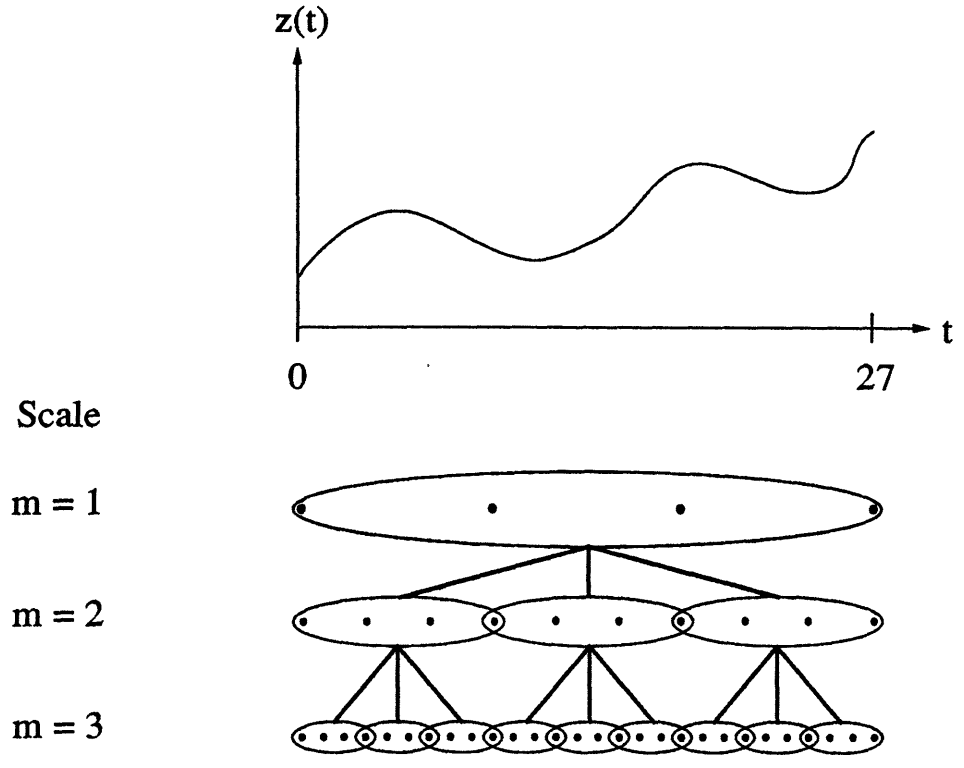


Figure 8: The state vectors are shown for a multiscale representation on a *third-order* tree.

Gibb's sampler is simple. 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. In general, this may not be a straightforward task, since, if we begin with a Gibb's distribution the computation of such pdf's requires explicit calculation of quantities related to the so-called *partition function* [33], which can be quite complex. On the other hand, such calculations need only be done once to construct the pdf's needed for the multiscale model. In addition, as we have seen for Brownian motion and as we now illustrate further, in many cases, including all vector Gauss-Markov processes, closed form expressions can be derived for the multiscale representations.

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, in particular describe this construction in detail for two cases corresponding to the integral of white noise (i.e. Brownian motion) and the second integral of white noise. These examples allow us to illustrate the interpretation of these multiresolution models as providing approximations using non-orthogonal expansions. In particular, our model for Brownian motion corresponds to the use of the so-

called "hat" function [57] in this expansion, leading to linear interpolation between dyadic points, while the model for the integral of Brownian motion leads to a multiresolution approximation using *cubic* interpolation.

The second part of this section presents several simple discrete-state examples, the first of which investigates general N-state Markov chains and allows us to make contact with the models used in [7, 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. In particular, through this example we gain insight into the very particular conditions that the parent-to-child transition pdf's must satisfy in order for the finest level process to be Markov. This analysis suggests that Markov chains are, in fact, only a small subset of the processes realizable with multiscale models and, in particular, directly motivates several other multiscale mid-point deflection processes which are not Markov.

3.3.1 Gauss-Markov Processes

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

$$\dot{z}(t) = F_t z(t) + G_t \mu(t) \tag{23}$$

where:

$$z(0) \sim \mathcal{N}(0, \Pi_0) \tag{24}$$

$$\mathbf{E}\{\mu(t)\mu(\tau)^T\} = I\,\delta(t-\tau) \tag{25}$$

$$\mathbf{E}\{\mu(t)z(0)^T\} = 0 \tag{26}$$

Define the state transition matrix by:

$$\dot{\Phi}(t,\tau) = F_t \Phi(t,\tau) \tag{27}$$

$$\Phi(t,t) = I \tag{28}$$

and state covariance matrix $\Pi_t = \mathbf{E}\{z(t)z(t)^T\}$. As is well known [27], the state covariance matrix satisfies the following differential and integral equations:

$$\dot{\Pi}_t = F_t \Pi_t + \Pi_t F_t^T + G_t G_t^T \tag{29}$$

$$\Pi_t = \Phi(t,0)\Pi_0\Phi(t,0)^T + \int_0^t \Phi(t,\tau)B_\tau B_\tau^T \Phi(t,\tau)^T d\tau \tag{30}$$

Also, define the conditional expectation of the state at time t_2 given the states $z(t_1)$ and $z(t_3)$, and the corresponding covariance of the error as:

$$\hat{z}_{t_2|t_1,t_3} = \mathbf{E}\{z(t_2)|z(t_1), z(t_3)\} \tag{31}$$

$$P_{t_2|t_1,t_3} = \mathbf{E}\{(z(t_2) - \hat{z}_{t_2|t_1,t_3})(z(t_2) - \hat{z}_{t_2|t_1,t_3})^T\} \tag{32}$$

Since:

$$p_{z(t_2)|z(t_1),z(t_3)}(Z_{t_2}|Z_{t_1}, Z_{t_3}) = \mathcal{N}(Z_{t_2}; \hat{z}_{t_2|t_1,t_3}, P_{t_2|t_1,t_3}) \tag{33}$$

the conditional expectation (31) and error covariance (32) are the quantities we require to construct a multiscale representation of the process (23) - (26). In fact, it is easy to show that for $t_1 < t_2 < t_3$:

$$\hat{z}_{t_2|t_1,t_3} = \begin{bmatrix} \Pi_{t_1}\Phi(t_2,t_1)^T \\ \Phi(t_3,t_2)\Pi_{t_2} \end{bmatrix}^T \begin{bmatrix} \Pi_{t_1} & \Pi_{t_1}\Phi(t_3,t_1)^T \\ \Phi(t_3,t_1)\Pi_{t_1} & \Pi_{t_3} \end{bmatrix}^{-1} \begin{bmatrix} z(t_1) \\ z(t_3) \end{bmatrix} \tag{34}$$

$$P_{t_2|t_1,t_3} = \Pi_{t_2} - \begin{bmatrix} \Pi_{t_1}\Phi(t_2,t_1)^T \\ \Phi(t_3,t_2)\Pi_{t_2} \end{bmatrix}^T \begin{bmatrix} \Pi_{t_1} & \Pi_{t_1}\Phi(t_3,t_1)^T \\ \Phi(t_3,t_1)\Pi_{t_1} & \Pi_{t_3} \end{bmatrix}^{-1} \begin{bmatrix} \Pi_{t_1}\Phi(t_2,t_1)^T \\ \Phi(t_3,t_2)\Pi_{t_2} \end{bmatrix} \tag{35}$$

Equations (34) and (35) directly provide us with the parameters of the matrices $A(s)$, $B(s)$ and P_0 in the multiscale model (1) - (3) corresponding to our mid-point deflection construction. In particular, 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:

$$x(s) \equiv x((m,\varphi)) = \begin{bmatrix} z(2\varphi/2^m) \\ z((2\varphi+1)/2^m) \\ z((2\varphi+2)/2^m) \end{bmatrix} \tag{36}$$

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

$$A(s) \equiv A((m,\varphi)) = \begin{cases} \begin{bmatrix} I & 0 & 0 \\ K_1 & K_2 & 0 \\ 0 & I & 0 \end{bmatrix} & \text{if } \varphi \text{ is even} \\ \begin{bmatrix} 0 & I & 0 \\ 0 & K_1 & K_2 \\ 0 & 0 & I \end{bmatrix} & \text{if } \varphi \text{ is odd} \end{cases} \tag{37}$$

In particular, if φ is even, then the first and third components of the state $x(s)$ in (36) correspond to the *first and second* components of $x(s\bar{\gamma})$. Thus, the identity matrices in (37) 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 (37). 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 (37) can be computed directly from (34). Using standard formulas for the inversion of a block 2×2 matrix, we compute:

$$\begin{aligned} K_1 = {} & \Phi(t_2,t_1) + \Phi(t_2,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T(\Pi_{t_3} - \Phi(t_3,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T)^{-1}\Phi(t_3,t_1) \\ & -\Pi_{t_2}\Phi(t_3,t_2)^T(\Pi_{t_3} - \Phi(t_3,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T)^{-1}\Phi(t_3,t_1) \end{aligned} \tag{38}$$

$$K_2 = -\Phi(t_2,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T(\Pi_{t_3} - \Phi(t_3,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T)^{-1} + \Pi_{t_2}\Phi(t_3,t_2)^T(\Pi_{t_3} - \Phi(t_3,t_1)\Pi_{t_1}\Phi(t_3,t_1)^T)^{-1} \quad (39)$$

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

The matrix $B(s)$ in (1) has the following block structure:

$$B(s) \equiv B((m,\varphi)) = \begin{bmatrix} 0 \\ K_3 \\ 0 \end{bmatrix} \quad (40)$$

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 (40) 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 error (35) is added to the second. In particular, from (3) we see that in this case the covariance of the additive noise term $B(s)w(s)$ in (1) is given by:

$$\mathbf{E}\{B(s)w(s)w^T(s)B^T(s)\} = B(s)B^T(s) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_{(2\varphi+1)/2^m|2\varphi/2^m,(2\varphi+2)/2^m} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (41)$$

where $s \equiv (m,\varphi)$.

Finally, the initial covariance matrix P_0 in (2) is given by:

$$P_0 \equiv \mathbf{E}\left\{ \begin{bmatrix} z(0) \\ z(0.5) \\ z(1) \end{bmatrix} \begin{bmatrix} z(0) \\ z(0.5) \\ z(1) \end{bmatrix}^T \right\} \quad (42)$$

$$= \begin{bmatrix} \Pi_0 & \Pi_0\Phi(1/2,0)^T & \Pi_0\Phi(1,0)^T \\ \Phi(1/2,0)\Pi_0 & \Pi_{1/2} & \Pi_{1/2}\Phi(1,1/2)^T \\ \Phi(1,0)\Pi_0 & \Phi(1,1/2)\Pi_{1/2} & \Pi_1 \end{bmatrix} \quad (43)$$

For instance, if $z(t)$ is the standard Brownian motion, then $F_t = 0$, $\Phi(t,\tau) = 1$, and $\Pi_t = t$. Thus for this example (34) and (35) become:

$$\hat{z}_{t_2|t_1,t_3} = \frac{t_3 - t_2}{t_3 - t_1}z(t_1) + \frac{t_2 - t_1}{t_3 - t_1}z(t_2) \quad (44)$$

$$P_{t_2|t_1,t_3} = \frac{(t_2 - t_1)(t_3 - t_2)}{t_3 - t_1} \quad (45)$$

and we obtain:

$$A(s) \equiv A((m,\varphi)) = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \text{if } \varphi \text{ is even} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix} & \text{if } \varphi \text{ is odd} \end{cases} \quad (46)$$

$$B(s) \equiv B((m,\varphi)) = \begin{bmatrix} 0 \\ 1/2^{(m+1)/2} \\ 0 \end{bmatrix} \tag{47}$$

$$P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix} \tag{48}$$

The formula (34) for 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 provides us with a direct interpretation of this representation as providing a sequence of multiresolution approximations. For example, Brownian motion provides us with the linear interpolation formula given in (44) and illustrated in Figure 1. This corresponds to a multiresolution linear spline approximation or, as also illustrated in Figure 1, as a non-orthogonal multiresolution decomposition using the so-called "hat" function [57].

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:

$$\dot{z}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu(t) \tag{49}$$

In (49), 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 are given by:

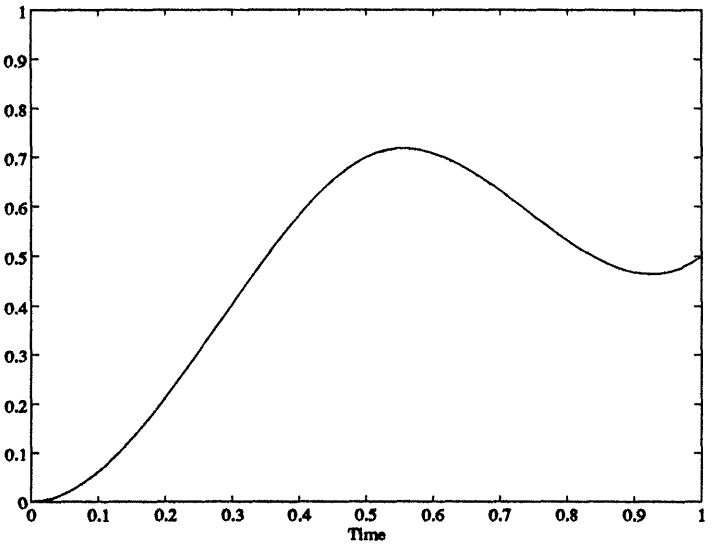
$$\Phi(t,\tau) = \begin{bmatrix} 1 & t-\tau \\ 0 & 1 \end{bmatrix} \tag{50}$$

$$\Pi_t = \begin{bmatrix} t^3/3 & t^2/2 \\ t^2/2 & t \end{bmatrix} \tag{51}$$

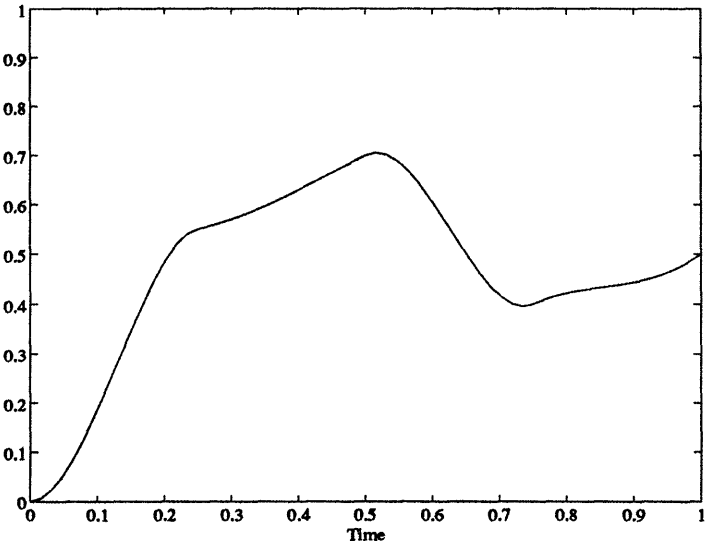
One can show by direct computation that the terms $\Pi_{t_1}\Phi(t_2,t_1)^T$ and $\Phi(t_3,t_2)\Pi_{t_2}$ in the leftmost block matrix on the right side of (34) contain only cubic powers of t_2. Note also that the block matrix in the middle of the right side does not depend on t_2. Thus, the interpolation of $z(t_2)$ between t_1 and t_3 is a cubic polynomial in t_2:

$$\hat{z}_{t_2|t_1,t_3} = \begin{bmatrix} c_1 + c_2 t_2 + c_3 t_2^2 + c_4 t_2^3 \\ c_2 + 2c_3 t_2 + 3c_4 t_2^2 \end{bmatrix} \tag{52}$$

where from (49), the second component of $\hat{z}_{t_2|t_1,t_3}$ is just the first derivative of the first. One can use (34) directly to calculate the values of the coefficients in (52). Alternatively, it is clear from the definition of $\hat{z}_{t_2|t_1,t_3}$ in (31) 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 (52), and thus uniquely determine the interpolating function (52). Note that the interpolating polynomial for the first component of the state (the position of the particle) 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 Figures 9a and 9b for the first two levels of a sample



a



b

Figure 9: 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 lower two graphs of Figure 1, which depict the piecewise linear interpolation of the *first* integral of white noise.

path of the multiscale realization. Figure 9a is the cubic interpolation of $z(0), z(0.5)$ and $z(1)$, while Figure 9b is the cubic interpolation of $z(0), z(0.25), z(0.5), z(0.75)$ and $z(1)$.

Finally, while we have focused 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:

$$z(t+1) = F_t z(t) + G_t \mu(t) \tag{53}$$

Also, as discussed in Section 3.2, in this case we can either construct models with three-point state vectors (e.g. $[z(0), z(T/2), z(T)]$ or four-point state vectors $[z(1), z(T/2), z(T/2+1), z(T)]$). The former of these is exactly analogous to what we have done here in continuous-time and has, as we have indicated, a high degree of redundancy, while the latter does not. We defer explicit discussion and illustration of such non-redundant representations until Section 4 where we describe them in the context of modeling 2-D MRF's.

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 conditions:

$$\Pr[z(0) = k] \tag{54}$$

for $k \in \{1, 2, \cdots L\}$ and by the one-step transition probabilities:

$$P_{i,j} \equiv \Pr[z(t) = i | z(t-1) = j] \tag{55}$$

We define the one-step transition matrix:

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,L} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ P_{L,1} & P_{L,2} & \cdots & P_{L,L} \end{bmatrix} \tag{56}$$

Note that the multistep transition probabilities are given by powers of the matrix[10] P:

$$\Pr[z(t+\tau) = i | z(t) = j] = [P^\tau]_{i,j} \tag{57}$$

Using (57) and Bayes' rule it is straightforward to calculate that for $t_1 < t_2 < t_3$:

$$\Pr[z(t_2) = j | z(t_1) = i, z(t_3) = k] = \frac{[P^{t_3 - t_2}]_{k,j} [P^{t_2 - t_1}]_{j,i}}{[P^{t_3 - t_1}]_{k,i}} \tag{58}$$

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

$$\Pr[z(0) = i, z(T/2) = j, z(T) = k] = [P^{T/2}]_{k,j} [P^{T/2}]_{j,i} \Pr[z(0) = i] \tag{59}$$

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

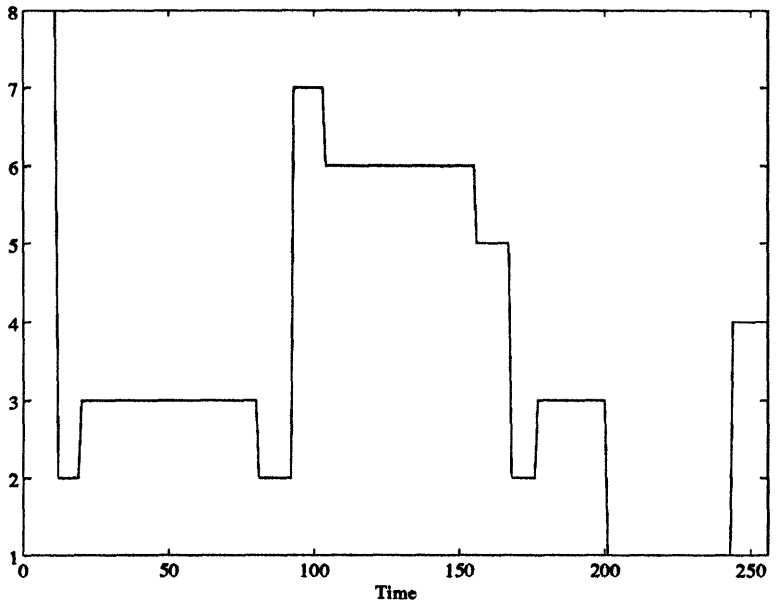


Figure 10: A sample path of a discrete-state Markov process is shown. A multiscale representation of this 8 state process is developed in Section 3.3.2.

allow us to construct the multiscale representation of the process. Note that (58) is the counterpart of the conditional probability equations for Gauss-Markov processes given in (33) – (35), and that the pdf for the state at the root node (59) is the counterpart of the initial covariance matrix (43).

One special case of this process is the following:

$$P_{i,i} = \mu \tag{60}$$

$$P_{i,j} = \frac{1-\mu}{L-1} \tag{61}$$

$$\Pr[z(0) = i] = 1/L, \quad i = 1, 2, \cdots, L \tag{62}$$

A sample path of such a process is shown in Figure 10 for $L = 8$ and $\mu = 0.97$. 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 (6) introduced in [7, 8, 9]. As noted in Section 2.2, the model in (6) does *not* in general produce a Markov chain or reciprocal process at the finest level. On the other hand (60) – (62) *is* a Markov model and for this process:

$$[P^k]_{i,j} = \begin{cases} (1 + (L-1)\vartheta^k)/L & \text{if } i = j \\ (1 - \vartheta^k)/L & \text{if } i \neq j \end{cases} \tag{63}$$

where:

$$\vartheta = (L\mu - 1)/(L - 1) \tag{64}$$

The conditional probability function (63) can be verified by noting that, given (60), (61), the one-step transition matrix (56) is circulant, and thus diagonalized by the $L \times L$ Discrete Fourier Transform matrix. Alternatively, by writing the one-step transition matrix for this example as the sum of a multiple of the identity, plus a matrix with constant entries, one can directly calculate the eigenvalues and eigenvectors, which again can be used to diagonalize (56).

Using (58), for this example we can write down the transition probabilities for the midpoint 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:

$$x(s) \equiv x((m,\varphi)) = \begin{bmatrix} z(2\varphi T/2^m) \\ z((2\varphi T + T)/2^m) \\ z((2\varphi T + 2T)/2^m) \end{bmatrix} \tag{65}$$

where, as in (36), 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:

$$\Pr[z(\frac{2\varphi T + T}{2^m}) = j | z(\frac{2\varphi T}{2^m}) = i, z(\frac{2\varphi T + 2T}{2^m}) = k] = \begin{cases} \zeta_1\zeta_1/\zeta_2 & \text{if } i = j = k \\ \zeta_1\xi_1/\xi_2 & \text{if } i \neq j = k \\ \zeta_1\xi_1/\xi_2 & \text{if } i = j \neq k \\ \xi_1\xi_1/\zeta_2 & \text{if } i = k \neq j \\ \xi_1\xi_1/\xi_2 & \text{if } i, j, k \text{ distinct} \end{cases} \tag{66}$$

where:

$$\zeta_1 = (1 + (L-1)\vartheta^{T/2^{m-1}})/L \tag{67}$$

$$\zeta_2 = (1 + (L-1)\vartheta^{T/2^{m-2}})/L \tag{68}$$

$$\xi_1 = (1 - \vartheta^{T/2^{m-1}})/L \tag{69}$$

$$\xi_2 = (1 - \vartheta^{T/2^{m-2}})/L \tag{70}$$

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:

$$P = \begin{bmatrix} 1-\mu & \eta \\ \mu & 1-\eta \end{bmatrix} \tag{71}$$

$$\Pr[z(0) = i] = \begin{cases} \eta/(\eta+\mu) & \text{if } i = 1 \\ \mu/(\eta+\mu) & \text{if } i = 2 \end{cases} \tag{72}$$

For this process one can show that:

$$P^k = \frac{1}{\eta+\mu} \begin{bmatrix} \eta + \mu(1-\eta-\mu)^k & \eta + \eta(1-\eta-\mu)^k \\ \mu + \mu(1-\eta-\mu)^k & \mu + \eta(1-\eta-\mu)^k \end{bmatrix} \tag{73}$$

and thus using (73) with (58), (59) one can build multiscale representations for the class of stationary binary Markov processes. While we have focused in this example and in the previous one on processes with stationary state transition probabilities, it is straightforward to apply this construction to the representation of non-stationary discrete-state Markov processes as well, simply by choosing the conditional probability functions required in the multiscale representation correctly.

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 $t \in \{0, 1, \cdots 2^N\}$ [59]:

$$\Pr[z(0) = i, z(2^{N-1}) = j, z(2^N) = k] = 1/8 \quad \text{for all } i, j, k \tag{74}$$

$$\Pr[z(t_2) = i | z(t_1) = j, z(t_3) = k] = \begin{cases} \mu & \text{if } i = j = k \\ 1 - \mu & \text{if } i \neq j \text{ and } j = k \\ 0.5 & \text{if } j \neq k \end{cases} \tag{75}$$

where $i, j, k \in \{1, 2\}$ and where t_1, t_2, t_3 are 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:

$$\Pr[z(t_2) = 1 | z(t_1) = 1, z(t_3) = 1] = \lambda_1 \tag{76}$$

$$\Pr[z(t_2) = 1 | z(t_1) = 1, z(t_3) = 2] = \lambda_2 \tag{77}$$

$$\Pr[z(t_2) = 1 | z(t_1) = 2, z(t_3) = 1] = \lambda_3 \tag{78}$$

$$\Pr[z(t_2) = 1 | z(t_1) = 2, z(t_3) = 2] = \lambda_4 \tag{79}$$

Since the process is represented with an N level multiscale process, there are $2^N - 2$ of these conditional densities which must be specified, corresponding to each of the nodes except the root node. However, the probability function for the state at the root node also requires four parameters, and thus the total number of parameters to be specified is $4(2^N - 1)$. 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 of these models 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 $4(2^N - 1)$-dimensional unit cube, while the set of non-stationary Markov chains over the same length interval $(2^N + 1)$ corresponds to the $2(2^N + 1)$-dimensional unit cube. Thus, for $N > 1$, Markov processes constitute only a "thin" subset of the entire class of binary tree processes.

4 Represention of 2-D Markov Random Fields

In this section we first review a few of the properties of Markov random fields and then describe how Markov random fields can be represented exactly with our multiscale modeling structure. Next, we introduce a family of approximate representations for Gaussian MRF's employing 1-D wavelet transforms.

4.1 2-D Markov Random Fields

Markov random fields (MRF's) are a multidimensional generalization of 1-D reciprocal processes. A continuous space stochastic process $z(t), t \in \mathcal{R}^n$ is said to be a Markov random field if the probability distribution of the process in the interior of any closed set Ω is independent of the process outside, conditioned on the values of $z(t)$ on the boundary Γ of Ω. That is, for $t \in \Omega \setminus \Gamma$:

$$p_{z(t)|z(\tau),\tau\in(\Omega\setminus\Gamma)^c}(Z_t|Z_\tau, \tau \in (\Omega \setminus \Gamma)^c) = p_{z(t)|z(\tau),\tau\in\Gamma}(Z_t|Z_\tau, \tau \in \Gamma) \tag{80}$$

where the notation $\Omega \setminus \Gamma$ denotes the set of elements in Ω which are not in Γ (in this case, the interior of Ω). The definition for Markov random fields on discrete lattices requires the specification of the notion of the "boundary" of a set in $\mathcal{Z}^n$ [64, 29]. Typically, this is accomplished through the specification of a *neighborhood system*. The idea is that the probability distribution of $z(t)$, conditioned on a set $\{z(\tau), \tau \in D_t\}$ in the neighborhood, D_t, of t, is independent of the process outside the neighborhood:

$$p_{z(t)|z(\tau),\tau\in\mathcal{Z}^n\setminus\{t\}}(Z_t|Z_\tau, \tau \in \mathcal{Z}^n \setminus \{t\}) = p_{z(t)|z(\tau),\tau\in D_t}(Z_t|Z_\tau, \tau \in D_t) \tag{81}$$

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 [13]. 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. The sequence of neighborhoods up to order seven is illustrated in Figure 11.

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:

$$\Gamma = \{\tau | \tau \in D_t, t \in \Omega\} \setminus \Omega \tag{82}$$

The conditional distribution and neighborhood structure cannot be chosen arbitrarily if one is to obtain a consistent joint distribution for the elements of the random field.

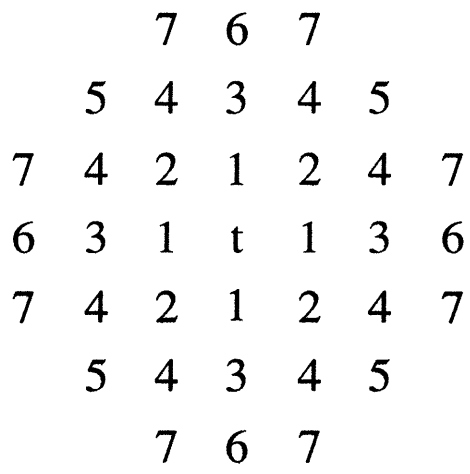


Figure 11: The first through seventh-order neighborhoods of lattice site t are shown. The first-order neighborhood consists of just the two vertical and two horizontal nearest neighbors.

First, the neighborhood system must have the properties that $t \notin D_t$ and (2) if $t \in D_\tau$ then $\tau \in D_t$. Second, the conditional distribution functions must satisfy the consistency conditions given by the Hammersley-Clifford theorem [4]. For detailed accounts of these issues and MRF's in general, we refer the reader to a few of the widely referenced papers in the field [30, 54, 56, 4, 64, 36, 33, 29].

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 Markov random fields on a square lattice. 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 *exactly* as in (1) – (3) but where s denotes a node on the quadtree depicted in Figure 12 and $\bar{\gamma}$ is a four-to-one operator, i.e. each point is the parent of four descendant points at the next level.

Quadtree structures arise naturally in multiresolution image processing contexts. For instance, successive filtering and decimation operations lead to images defined on such a hierarchy of grids in the Laplacian pyramid coding algorithm of Burt and Adelson [11] and in the closely related wavelet transform decomposition of images [44]. Also, the multigrid approaches to low level vision problems discussed by Terzopoulos [60] involve relaxation on a similar sequence of grids. In addition, Wilson and his colleagues [21] have introduced and used a particular simple example of the model in (1) – (3) with the state defined on a quadtree for image processing.

In fact, it is interesting to note that both in [21] and in the segmentation studies in [7, 8, 9] a similar deficiency in the quadtree models was discussed. In particular, processing based on these models led to blocky results which could differ significantly for different

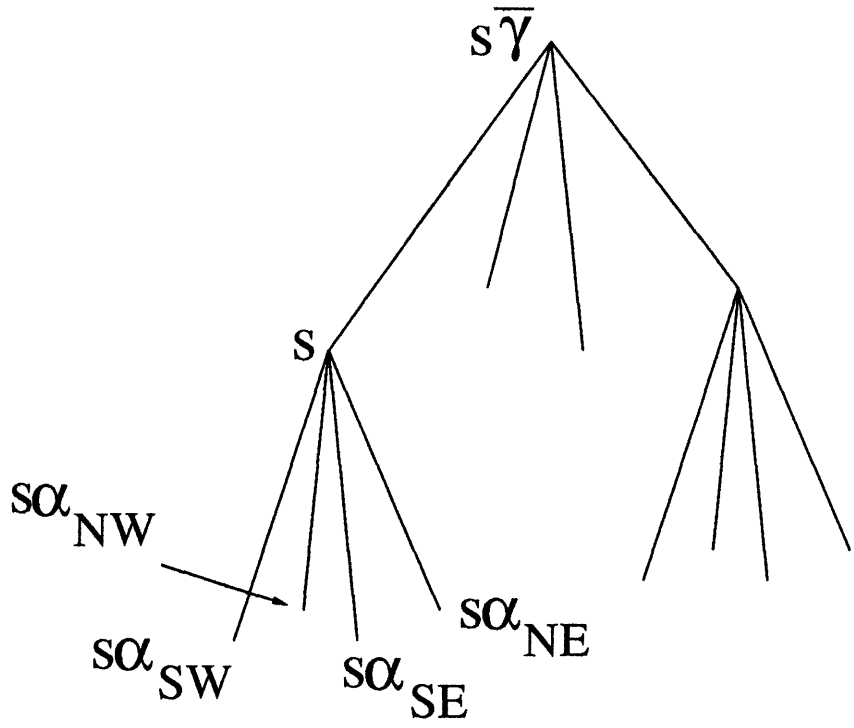


Figure 12: The quadtree structure shown is used for the multiscale representations of Markov random fields (MRF's). Each node of the quadtree has four offspring, denoted $s\alpha_{NW}, s\alpha_{NE}, s\alpha_{SE}, s\alpha_{SW}$. Again, the parent of node s is denoted $s\bar{\gamma}$, and in this case $\bar{\gamma}$ is a *four-to-one* shift operator.

positionings of the tree with respect to the finest scale $\mathcal{Z}^2$ lattice, and this fact led to the need for more complex processing structures. In particular, in [21] it was necessary to perform processing several times with different tree positions and to average the results, while in [7, 8, 9] the tree was replaced by a more connected lattice (so that any two nodes have both common ancestors *and* common descendants). The latter modification, however, destroys the partially-ordered Markovian structure which the dyadic and quadtree processes possess and which leads to highly parallelizable and scale-*recursive*, rather than iterative, algorithms. In this section, we show that one can completely avoid the apparent problems in using quadtree models by demonstrating that one can model *any* MRF *exactly* using such a model.

Consider 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. 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 13.

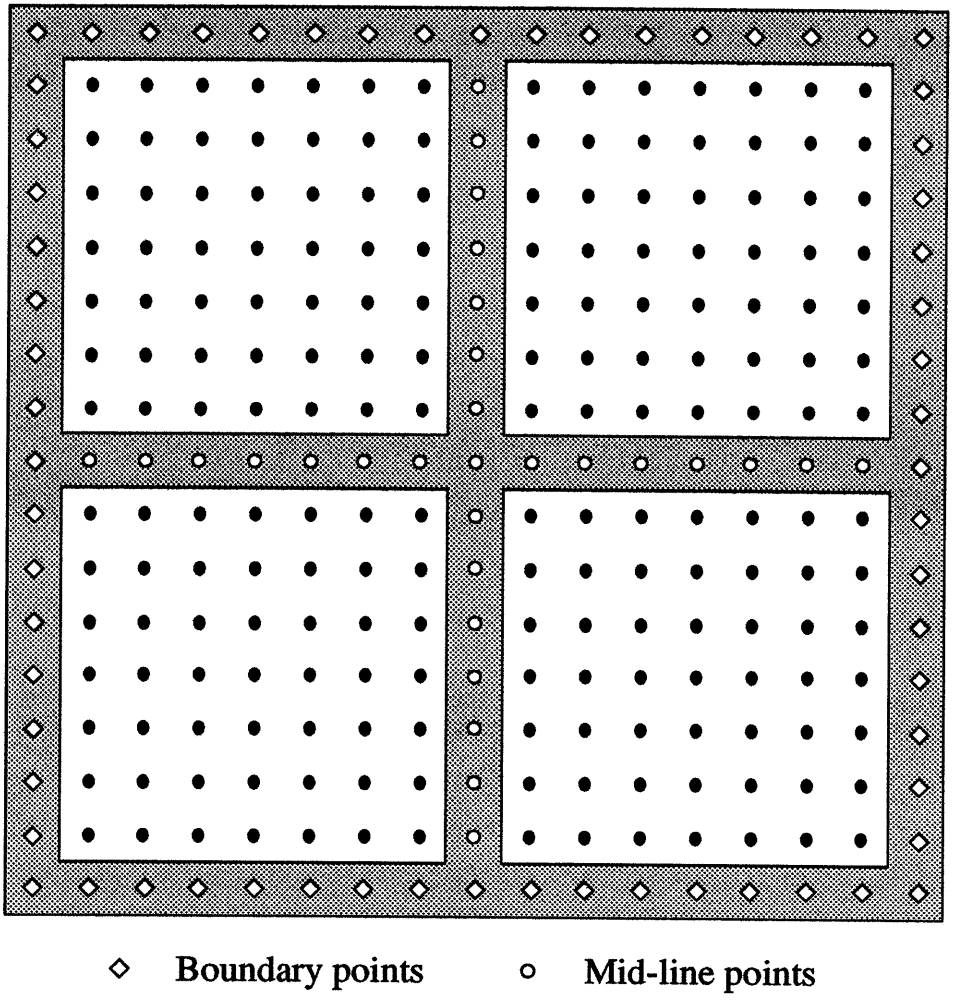


Figure 13: The state vector at the root node in the MRF multiscale representation consists of the MRF values at the boundary and "mid-line" points, shown in the shaded region here for a 17×17 lattice. To construct a sample path of the MRF using the "mid-line" deflection construction, we start by choosing a sample from the joint distribution of the values in the root node state.

The boundary points are denoted with $\diamond$ and o 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 in (20) when constructing a 1-D reciprocal process.

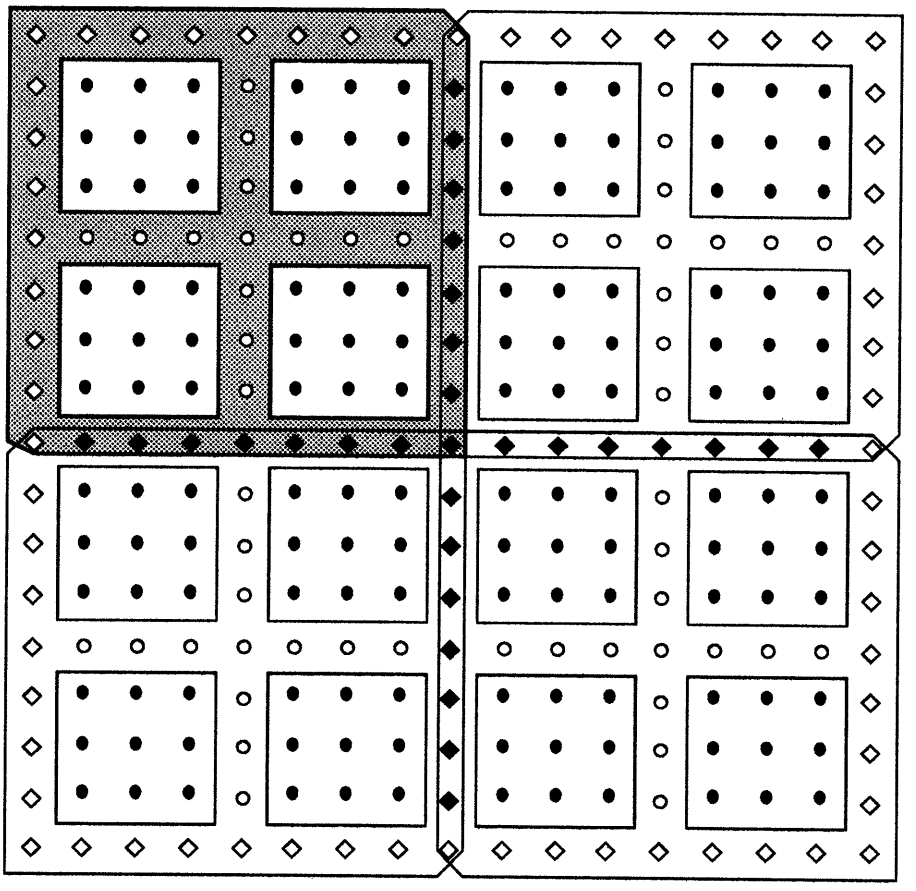
In the 1-D case, transitions from the first to second level consisted of obtaining a sample from the conditional distribution of the state at the mid-points of the left and right half-intervals. In two dimensions, we predict the set of values at the mid-lines in each of the

four quadrants. The components of the four state vectors at the second level are illustrated in Figure 14 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 3). In general, the state vectors at the four nodes at the second level each specify the MRF at $6 \times 2^{N-1} - 3$ lattice points. Each of these 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 14) as well as new mid-line points within each quadrant (the open circles in Figure 14). 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 (21), (22) 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) – (3).

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. 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 finest scale.

Third, note that the representation we have defined is redundant. In the 1-D case this redundancy was removed by predicting two mid-points at each level instead of one (cf.



◇ **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 14: 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 6). Likewise, in two dimensions we can eliminate the redundancy by predicting *two* mid-lines at each level instead of one. For instance, the state vector at the top level of the quadtree for an MRF defined on a 16×16 lattice would consist of the values of the MRF at the shaded points shown in Figure 15, and the four state vectors at the next level would consist of the values of the process at the points shaded in Figure 16. In the 2-D case this latter representation is important in the context of defining approximate models, as will be seen in the next section, and illustrated in Section 5.

Fourth, higher-order models can be easily accommodated. For example, if the Markov random field has up to a fifth-order neighborhood (cf. Figure 11), the field can be represented by taking as state the values of the process along boundaries and mid-lines of "width" two. In general, neighborhoods of any order can be handled by increasing the boundary and mid-line widths appropriately.

Fifth, in the case of Gaussian MRF's, the prediction from one level to the next in our representation, as captured by the term $A(s)x(s\bar{\gamma})$ in (1), is simply the result of a 2-D version of the interpolation formula displayed in (31) for general 1-D Gauss-Markov processes and in particular in (44) for Brownian motion. As in 1-D, this interpolation formula can also be thought of as providing a multiresolution approximation to a process using interpolation functions specifically tailored to the process under consideration. 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:

$$\frac{d^2}{dt^2}\hat{z}_{t|0,T} = 0 \tag{83}$$

Similarly the interpolation of the first component of the second-order process (49) is given by the solution of:

$$\frac{d^4}{dt^4}\hat{z}_{t|0,T} = 0 \tag{84}$$

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:

$$\nabla^2 \hat{z} = 0 \tag{85}$$

given values of z on the boundary of a square region, while the counterpart to (84), corresponding to a second-order model, would be the solution of a homogeneous biharmonic equation:

$$\nabla^4 \hat{z} = 0 \tag{86}$$

given boundary values and normal derivatives along the boundary. More generally, the particular interpolation formula for a 1-D or 2-D process is given by the solution of a specific homogeneous differential (or partial differential) equation determined by the covariance structure of the process (see, for example [63, 39], for related discussions).

Finally, we note that for domains of substantial size, the representations may be of prohibitively large dimension. This issue is addressed for Gaussian MRF's in the next section, where we introduce a family of low-dimensional *approximate* representations based on *one-dimensional* wavelet transforms of the MRF along 1-D boundaries.

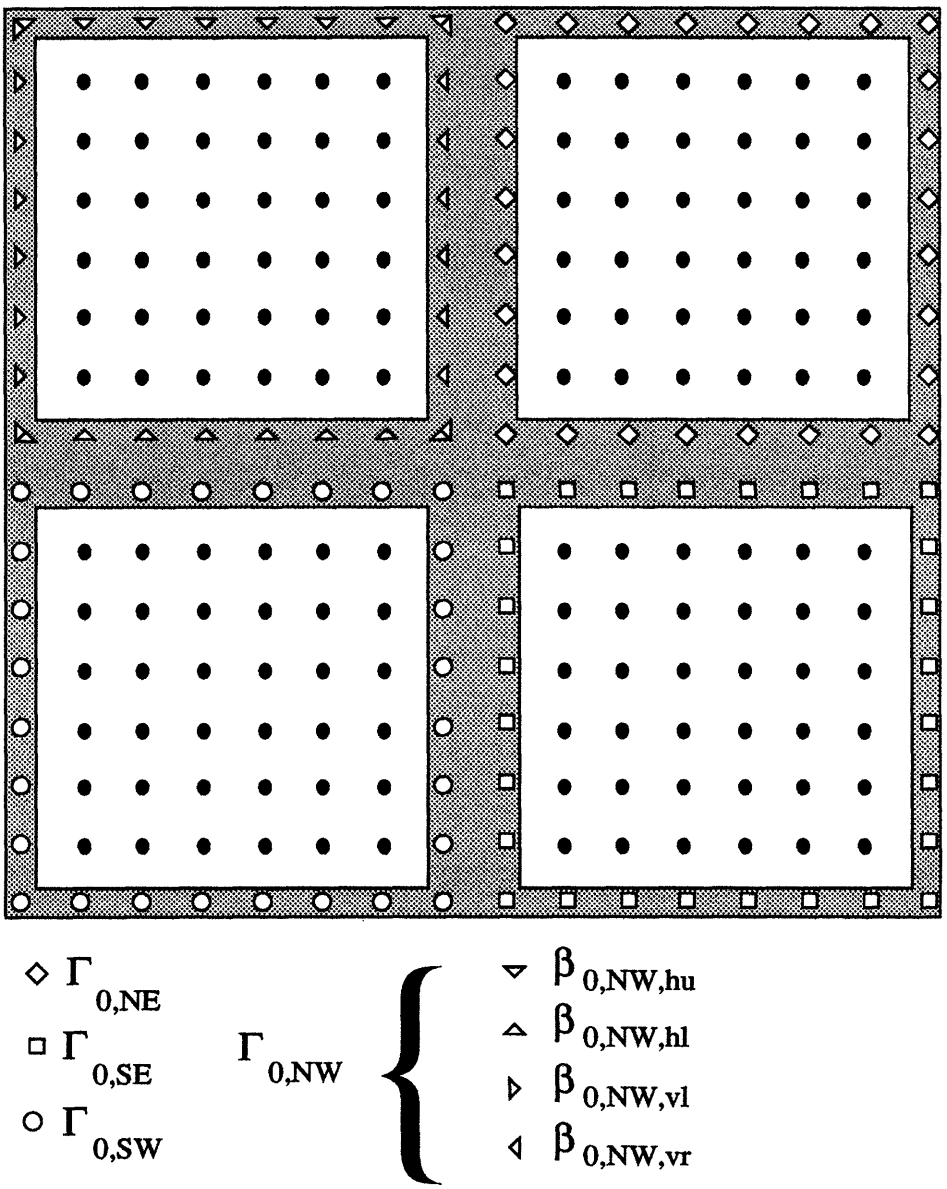


Figure 15: The state at the root node in a *non-redundant* exact multiscale representation of an MRF defined on a 16 × 16 lattice consists of the values of the process at the shaded points. The redundancy in the exact representation is eliminated by generating the values of the process along two mid-lines 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. $\triangledown$ 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.

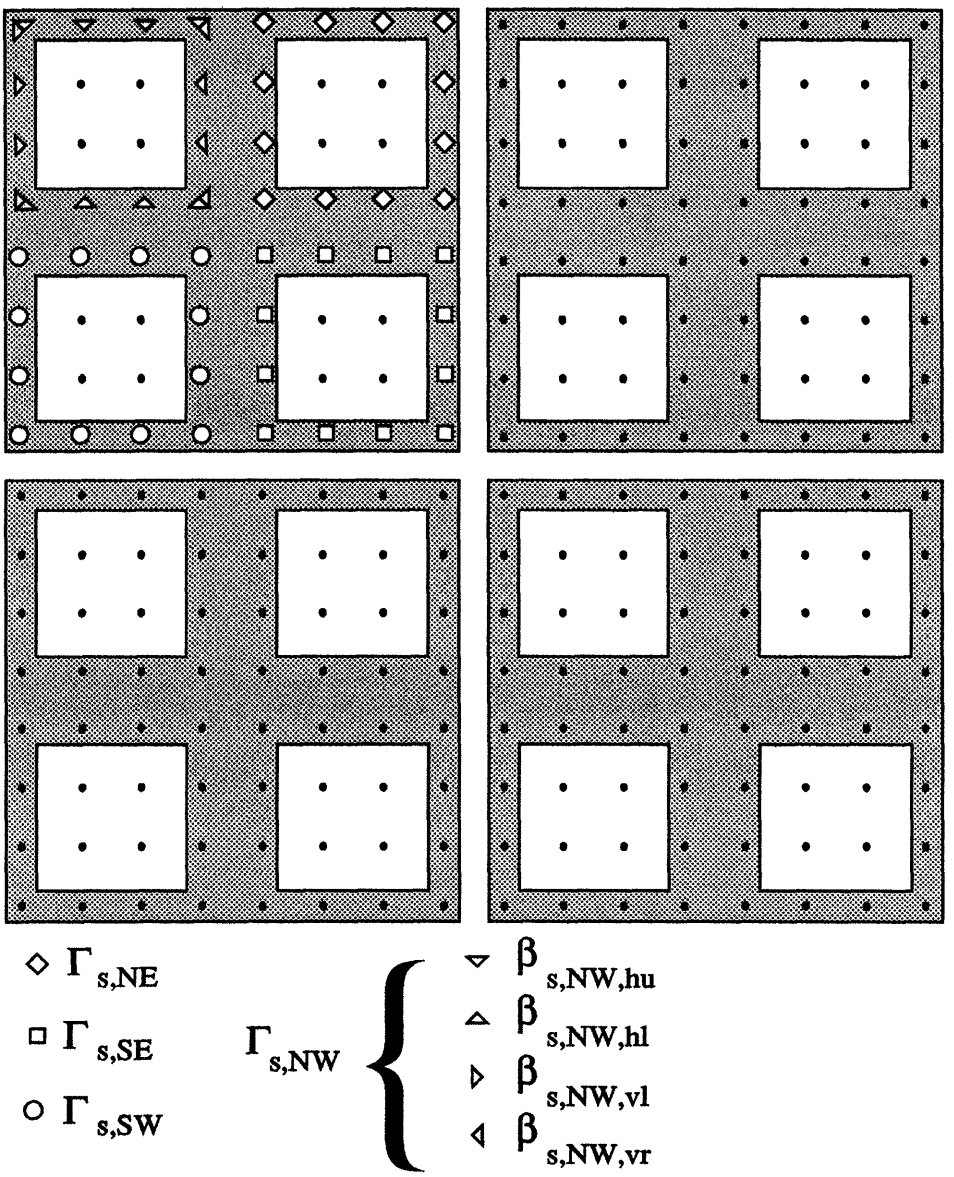


Figure 16: The four states at the second level of the tree in a non-redundant exact multiscale representation are scaled and shifted versions of the state at the root node, and are shown here for an MRF defined on a 16×16 lattice. The state in the north-west corner contains the values of the process 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.

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 the exact multiscale representations. The basic 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 a Gaussian 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 imagine using a wide variety of different coordinate transformations, including a 1-D Fourier transform or a Karhunen-Loève expansion, we focus here on a choice that is particularly well matched to the self-similar, multiresolution nature of our exact representation. Specifically, as illustrated in Figures 15 and 16, as we proceed from one level to the next finer level in our multiscale representation of a Gaussian MRF, the boundaries at the coarser levels are essentially halved in length (and new, shorter boundaries are added as well). This self-similar structure suggests representing the values of the MRF along such boundaries in terms of wavelet bases. In this case, if at each level we keep only the wavelet coefficients up to a particular level of resolution, then at each level in our representation we are actually adding only one new level of detail.

The approximation that is made in keeping only coarse wavelet approximations along boundaries actually has two parts. The first is that we assume that the new detail to be added along each boundary from level to level is independent of the previously generated coarse approximations. The second is that we neglect the residual correlation between MRF values in neighboring subregions when we are given only a coarse approximation, rather than complete knowledge of, the values along their common boundary. The former of these approximations, namely the scale-to-scale decorrelation capability of wavelet transforms, has already been studied and well documented in several papers on 1-D stochastic processes [19, 61]. The latter of these, which deals explicitly with the full statistical structure of 2-D MRF's, has not, to our knowledge, been investigated previously (in fact, the use of 1-D wavelets for 2-D random fields is, we believe, a completely new idea). As the results presented here illustrate, this appears to be an extremely effective method for many MRF's.

The approximate models are based on the *non-redundant* exact representations for MRF's described in the previous section. The states at the first and second levels of this representation for an MRF defined on a 16×16 lattice are shown in Figures 15 and 16. The root node state in Figure 15 contains the values of the MRF at 112 points. More generally, in a multiscale representation of an MRF defined on a $2^N \times 2^N$ lattice, a state at

the m^{th} level in this exact representation 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 again the 112 points associated with the root node state in Figure 15. 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 the set Γ_s into four sets of $4(2^{N-m(s)}-1)$ points in a similar fashion, and we denote these mutually exclusive subsets as $\Gamma_{s,i}, i \in \{NW, NE, SE, SW\}$, where the subscripts refer to the spatial location of the subset. For instance, with $s = 0$ corresponding to the root node, the four subsets $\Gamma_{0,i}, i \in \{NW, NE, SE, SW\}$ are illustrated in Figure 15 with the symbols:

$$\Gamma_{0,NW} \leftrightarrow \triangledown, \triangleleft, \triangleright, \triangle, \text{ and combinations of these.} \tag{87}$$

$$\Gamma_{0,NE} \leftrightarrow \diamond \tag{88}$$

$$\Gamma_{0,SE} \leftrightarrow \square \tag{89}$$

$$\Gamma_{0,SW} \leftrightarrow \circ \tag{90}$$

$$\tag{91}$$

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 15 and defined below:[11]

$$\beta_{0,NW,hu}(k) = z(0,k), \quad \begin{array}{l}\text{corresponding to the points denoted with } \triangledown, \\ \text{the combination of } \triangledown \text{ and } \triangleleft, \\ \text{and the combination of } \triangledown \text{ and } \triangleright \text{ in Fig. 15.}\end{array} \tag{92}$$

$$\beta_{0,NW,vr}(k) = z(k,7), \quad \begin{array}{l}\text{corresponding to the points denoted with } \triangleleft, \\ \text{the combination of } \triangleleft \text{ and } \triangledown, \\ \text{and the combination of } \triangleleft \text{ and } \triangle \text{ in Fig. 15.}\end{array} \tag{93}$$

$$\beta_{0,NW,hl}(k) = z(7,k), \quad \begin{array}{l}\text{corresponding to the points denoted with } \triangle, \\ \text{the combination of } \triangle \text{ and } \triangleleft, \\ \text{and the combination of } \triangle \text{ and } \triangleright \text{ in Fig. 15.}\end{array} \tag{94}$$

$$\beta_{0,NW,vl}(k) = z(k,0), \quad \begin{array}{l}\text{corresponding to the points denoted with } \triangleright, \\ \text{the combination of } \triangleright \text{ and } \triangledown, \\ \text{and the combination of } \triangleright \text{ and } \triangle \text{ in Fig. 15.}\end{array} \tag{95}$$

[11] We will use $z(i,j)$ to denote the value of the MRF at lattice site (i,j). If the lattice has T_1 rows and T_2 columns, then $(i,j) \in \{0,1,\cdots,T_1-1\} \times \{0,1,\cdots,T_2-1\}$.

$$\beta_{0,NE,hu}(k) = z(0,k+8) \quad (96)$$
$$\beta_{0,NE,vr}(k) = z(k,15) \quad (97)$$
$$\beta_{0,NE,hl}(k) = z(7,k+8) \quad (98)$$
$$\beta_{0,NE,vl}(k) = z(k,8) \quad (99)$$
$$\beta_{0,SE,hu}(k) = z(8,k+8) \quad (100)$$
$$\beta_{0,SE,vr}(k) = z(k+8,15) \quad (101)$$
$$\beta_{0,SE,hl}(k) = z(15,k+8) \quad (102)$$
$$\beta_{0,SE,vl}(k) = z(k+8,8) \quad (103)$$
$$\beta_{0,SW,hu}(k) = z(8,k) \quad (104)$$
$$\beta_{0,SW,vr}(k) = z(k+8,7) \quad (105)$$
$$\beta_{0,SW,hl}(k) = z(15,k) \quad (106)$$
$$\beta_{0,SW,vl}(k) = z(k+8,0) \quad (107)$$

for $k = 0, 1, \cdots, 7$. 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 15 by the presence of both $\triangledown$ 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:

$$x(s) = \begin{bmatrix} x_{NW}(s) \\ x_{NE}(s) \\ x_{SE}(s) \\ x_{SW}(s) \end{bmatrix} \quad (108)$$

where:

$$x_i(s) = \begin{bmatrix} W_0\beta_{s,i,hu} \\ W_0\beta_{s,i,vr} \\ W_0\beta_{s,i,hl} \\ W_0\beta_{s,i,vl} \end{bmatrix} \quad (109)$$

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 (108),(109) (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 12). 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 nodes. In the *exact, non-redundant representations*, the following relationship holds:

$$p_{z(t),t\in\Gamma_{s\alpha_i}|z(\tau),\tau\in\Gamma_s}(Z_t, t \in \Gamma_{s\alpha_i}|Z_\tau, \tau \in \Gamma_s) = p_{z(t),t\in\Gamma_{s\alpha_i}|z(\tau),\tau\in\Gamma_{s,i}}(Z_t, t \in \Gamma_{s\alpha_i}|Z_\tau, \tau \in \Gamma_{s,i}) \quad (110)$$

for $i \in \{NW, NE, SE, SW\}$. What (110) 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 $x_{NE}(s), x_{SE}(s)$ and $x_{SW}(s)$, given $x_{NW}(s)$. 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 (108) – (109) corresponds to the coarsest of these. Using the same symbols $x_i(s)$ 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:

$$p_{x(s\alpha_i)|x(s)}(X_{s\alpha_i}|X_s) = p_{x(s\alpha_i)|x_i(s)}(X_{s\alpha_i}|X_i(s)) \tag{111}$$

Since the field being approximated is assumed to be jointly Gaussian, the conditional density function (111) is parameterized by conditional means and covariances as in (33) – (35):

$$p_{x(s\alpha_i)|x_i(s)}(X_{s\alpha_i}|X_i(s)) = \mathcal{N}(X_{s\alpha_i}; \hat{x}_{s\alpha_i}, P_{s\alpha_i}) \tag{112}$$

where:

$$\begin{aligned} \hat{x}_{s\alpha_i} &= \mathbf{E}\{x(s\alpha_i)|x_i(s)\} \\ &= \mathbf{E}\{x(s\alpha_i)x_i(s)^T\}(\mathbf{E}\{x_i(s)x_i(s)^T\})^{-1}X_i(s) \end{aligned} \tag{113}$$

$$\begin{aligned} P_{s\alpha_i} &= \mathbf{E}\{(x(s\alpha_i) - \hat{x}_{s\alpha_i})(x(s\alpha_i) - \hat{x}_{s\alpha_i})^T\} \\ &= \mathbf{E}\{x(s\alpha_i)x(s\alpha_i)^T\} \\ &\quad -\mathbf{E}\{x(s\alpha_i)x_i(s)^T\}(\mathbf{E}\{x_i(s)x_i(s)^T\})^{-1}(\mathbf{E}\{x(s\alpha_i)x_i(s)^T\})^T \end{aligned} \tag{114}$$

One can then derive the matrices $A(s), B(s)$ and P_0 in the multiscale representation of the random field:

$$A(s\alpha_{NW}) = [K_1, 0, 0, 0] \tag{115}$$

$$A(s\alpha_{NE}) = [0, K_2, 0, 0] \tag{116}$$

$$A(s\alpha_{SE}) = [0, 0, K_3, 0] \tag{117}$$

$$A(s\alpha_{SW}) = [0, 0, 0, K_4] \tag{118}$$

where:

$$K_i = \mathbf{E}\{x(s\alpha_i)(x_i(s))^T\}(\mathbf{E}\{x_i(s)(x_i(s))^T\})^{-1} \tag{119}$$

Also:

$$B(s\alpha_i)B(s\alpha_i)^T = P_{s\alpha_i} \quad (120)$$

$$P_0 = \mathbf{E}\{x_0 x_0^T\} \quad (121)$$

The assumption (111) is directly reflected in (115) - (118). In particular, the state $x(s\alpha_i)$ is a function *only* of the i^{th} component of the parent (cf. (108)). Thus, the assumption in (111) 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, $K_1 = K_2 = K_3 = K_4$. Thus, in this case, the representations are quite simply described, and more importantly, this simple structure, in addition to the substantially reduced dimensionality of the approximate representations, leads to considerable efficiencies for smoothing [17, 18, 19] and likelihood calculation algorithms [43].

As we have indicated, the generalization of the coarsest approximate model, with state given by (108), (109) 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 $2^{N-m(s)} - 1$ "detail" coefficients[12]. These are computed recursively according to[13]:

$$f_k^{j-1} = \sum_{n=1}^{n=2M} h_n f_{n+2k-2}^j \quad (122)$$

$$d_k^{j-1} = \sum_{n=1}^{n=2M} g_n f_{n+2k-2}^j \quad (123)$$

where the scaling coefficients and detail coefficients are f_k^j and d_k^j respectively, h_n, g_n are impulse responses of quadrature mirror filters [26, 55] 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

[12]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.

[13]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.

variables $x(s)$. In particular, simply replace (109) with:

$$x_i(s) = \begin{bmatrix} W_p\beta_{s,i,hu} \\ W_p\beta_{s,i,vr} \\ W_p\beta_{s,i,hl} \\ W_p\beta_{s,i,vl} \end{bmatrix} \tag{124}$$

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 (111), the parameters $A(s)$, $B(s)$ and P_0 are again given by (115) – (121).

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 the first or second-order neighborhood in Figure 11. 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 two or more, and utilizing these as the state. When the boundaries are expanded to have a width of q lattice sites, the state at node s will be broken up into the p^{th}-order representations of $16q$ sequences of length $2^{N-m(s)}$. 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. An example of an approximate representation which keeps wavelet coefficient along boundaries of width two is discussed in the next section.

Third, note that the covariance matrices required in (119) and (120) are not invertible if the representation of the 1-D boundary sequences is complete, due to the fact, as mentioned previously, that these sequences overlap. For instance, in Figure 15, both $\beta_{0,NW,hu}$ and $\beta_{0,NW,vl}$ contain the value of the state at pixel location $(0,0)$. In this case we have *redundant* information and hence the conditional expectation and error covariance formulas must be modified to deal with this. This modification is a straightforward matter as discussed in [50, 58].

Fourth, note that not only has the dimensionality of the representations 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.

Fifth, because our approximate models keep only limited resolution versions of MRF values along 1-D boundaries, the quadtrees for these approximate models may require more levels than the exact model. For example, consider an MRF over a 4×4 region. The exact representation of this field in our framework has only a single level, since the exterior boundaries and mid-lines form the *entire* 4×4 region (consider Figure 15 adapted to a 4×4 grid). On the other hand, a first-order Haar approximation would retain only the sixteen average values of pairs of horizontal or vertical points at the first level, only twelve of which are independent thanks to the overlap in the $\beta_{s,i,j}$ sequences. Consequently, in this case, we need a second level, corresponding to "averages" of single points, to completely represent the field.

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 several GMRF's and a number of different approximate representations.

5 Examples of Approximate 2-D Gaussian MRF Representations

In this section we illustrate sample paths generated by our approximate representations for two examples of separable Gaussian MRF's and then for two examples of non-separable Gaussian MRF's. Separable MRF's were one of the first widely used image models because of their simple covariance structure [34], while non-separable GMRF's have been widely used in the context of texture representation [13, 14, 22, 23, 24, 47, 48].

5.1 Separable Gaussian MRF Examples

Consider a separable Gaussian MRF defined on a $2^N \times 2^N$ lattice with a covariance function given by:

$$\mathbf{E}\{z(i,j)z(k,l)\} = \sigma^2 \rho_x^{|i-k|} \rho_y^{|j-l|} \tag{125}$$

where ρ_x and ρ_y are one-step correlation parameters in the vertical and horizontal directions. These random fields are Markovian with respect to the second-order neighborhood given in Figure 11.

Let us construct a zeroth-order Haar approximate representation of this MRF. In this case, the state at the root node of the tree consists of sixteen values, representing the averages across each of the 1-D components of the boundary. Since this state variable is just a linear function of the values of the random field, its covariance structure can be calculated directly from knowledge of the averages taken and the covariance function in (125). Indeed, the covariance matrix for the four averages corresponding to the north-west corner of the state at the root node is given by:

$$\mathbf{E}\left\{ \begin{bmatrix} W_0\beta_{0,NW,hu} \\ W_0\beta_{0,NW,vr} \\ W_0\beta_{0,NW,hl} \\ W_0\beta_{0,NW,vl} \end{bmatrix} \begin{bmatrix} W_0\beta_{0,NW,hu} \\ W_0\beta_{0,NW,vr} \\ W_0\beta_{0,NW,hl} \\ W_0\beta_{0,NW,vl} \end{bmatrix}^T \right\} = \begin{bmatrix} \omega_y & \psi & \varrho_x\omega_y & \psi \\ \psi & \omega_x & \psi & \varrho_y\omega_x \\ \varrho_x\omega_y & \psi & \omega_y & \psi \\ \psi & \varrho_y\omega_x & \psi & \omega_x \end{bmatrix} \tag{126}$$

where:

$$\omega_y = (2^{N-1}(1+\rho_y)/(1-\rho_y) - 2\rho_y(1-\rho_y^{2^{N-1}})/(1-\rho_y)^2 \quad (127)$$

$$\omega_x = (2^{N-1}(1+\rho_x)/(1-\rho_x) - 2\rho_x(1-\rho_x^{2^{N-1}})/(1-\rho_x)^2 \quad (128)$$

$$\psi = ((1-\rho_x^{2^{N-1}})/(1-\rho_x))((1-\rho_y^{2^{N-1}})/(1-\rho_y)) \quad (129)$$

$$\varrho_x = \rho_x^{2^{N-2}-1} \quad (130)$$

$$\varrho_y = \rho_y^{2^{N-2}-1} \quad (131)$$

Figures 17a and 17b illustrate 256×256 sample paths of exact representations of the separable random fields for $\rho_x = \rho_y = 0.7$ and $\rho_x = \rho_y = 0.9$, respectively. Sample paths of zeroth-order Haar approximations of these MRF's are shown in Figures 17c and 17d, respectively[14]. Note that for $\rho_x = \rho_y = 0.7$, the zeroth-order approximation is visually similar to the exact representation, with only minor boundary effects apparent, caused by the approximation in the first-order Haar model, i.e. the neglecting of the residual correlation in adjoining regions when we are given only the coarse Haar approximation of the field along common boundaries. As the coupling between pixels is increased, the effects of this coarse approximation become more apparent, as seen in the first-order approximation of the separable MRF with $\rho_x = \rho_y = 0.9$. This indicates that such fields will in general require higher-order approximations. We defer the illustration of such higher-order approximations of fields to the following subsection, in which we describe several examples of the use of our modeling methodology to represent natural textures.

5.2 Non-Separarable Gaussian MRF Examples

Consider the class of GMRF's defined by the following 2-D autoregressive model [12, 36]:

$$z(i,j) = \sum_{(k,l)\in D} r_{k,l} z(i-k, j-l) + e(i,j) \quad (132)$$

where $r_{k,l} = r_{-k,-l}$, D is a neighborhood around the origin $(0,0)$, the Gaussian driving noise $e(i,j)$ is a locally correlated sequence of random variables, and $(i,j) \in \{0,1,\cdots,2^N\} \times \{0,1,\cdots,2^N\}$. In the examples below, the set D corresponds to the neighborhood sets in Figure 11. For instance, the set corresponding to a first-order neighborhood is $D = \{(0,1),(0,-1),(1,0),(-1,0)\}$. In addition, we interpret the $2^N \times 2^N$ lattice as a *toroid*, i.e. the independent variables (i,j) in (132) are interpreted modulo 2^N. For instance, the first-order neighborhood of lattice site $(0,0)$ is given by the set $\{(1,0),(0,1),(0,2^N-1),(2^N-1,0)\}$. Finally, the correlation structure of the driving noise is given by:

$$\mathbf{E}\{e(i,j)e(i-k,j-l)\} = \begin{cases} \sigma^2 & \text{if } k=l=0 \\ -\sigma^2 r_{k,l} & \text{if } (k,l)\in D \\ 0 & \text{if } (k,l)\notin D \end{cases} \quad (133)$$

and has the property that:

$$\mathbf{E}\{e(i,j)z(k,l)\} = \begin{cases} \sigma^2 & \text{for } i=k, j=l \\ 0 & \text{else} \end{cases} \quad (134)$$

[14] We emphasize that the Figures 17c - 17d depict *sample paths* of the approximate representations, and *not* approximate representations of the sample paths in Figure 17a - 17b.

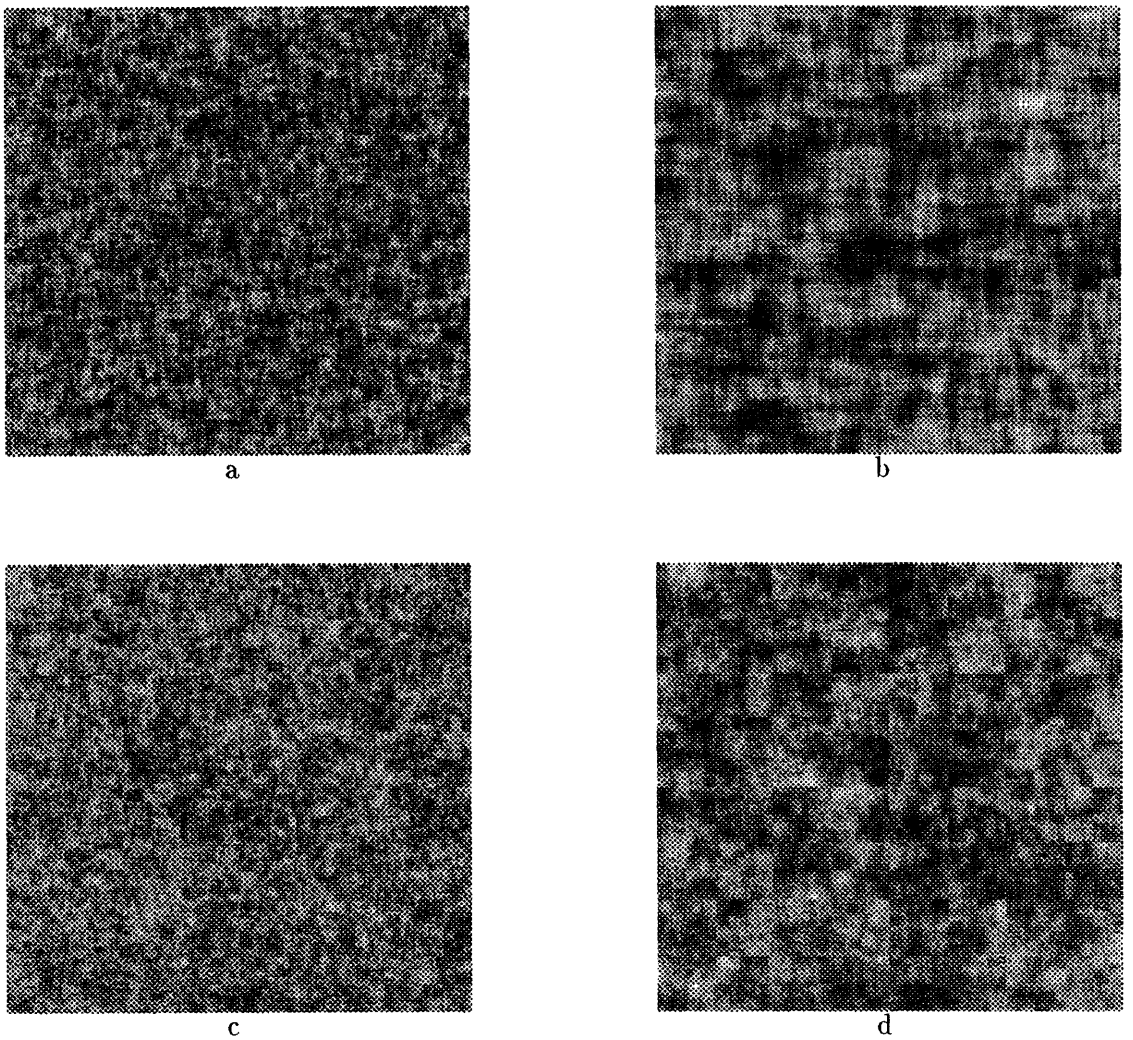


Figure 17: Sample paths of a separable MRF with correlation structure $\mathbf{E}\{z(i,j)z(k,l)\} = \sigma^2\rho_x^{|i-k|}\rho_y^{|j-l|}$ with $\rho_x = \rho_y = 0.7$ and $\rho_x = \rho_y = 0.9$ are shown in (a) and (b) respectively. Sample paths of zeroth-order approximate representations of these fields, based on the Haar wavelet, are shown in (c) and (d). The stronger correlation between neighboring pixels in (b) leads to boundary effects in the sample path of the approximate representation shown in (d), which can be eliminated by using higher-order approximations.

From (134), and the fact that the random field is Gaussian, one can prove that the autoregressive model above *does* imply that $z(i,j)$ is a Markov Random Field [64]. We refer to the model (132) as a Q^{th}-order MRF if the set D corresponds to the Q^{th}-order neighborhood of Figure 11.

Infinite lattice versions of the processes in (132) were introduced in [64] and their toroidal lattice counterparts have been thoroughly studied in the context of texture representation [13, 14, 22, 23, 24, 47, 48]. The correlation structure of these MRF's cannot be explicitly written down as in the previous example. However, the specific statistics and correlations (as in (115) – (118)) required to construct our multiscale approximate 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 [36]. In particular, denote by $\mathbf{z}$ the set of values $z(t)$ stacked into a vector (with lexicographic ordering), and denote the correlation matrix of the MRF by $R_{\mathbf{zz}}$. Then:

$$FR_{\mathbf{zz}}F^* = \Lambda \tag{135}$$

where F is the 2-D Fourier transform matrix and Λ is a diagonal matrix of the eigenvalues of $R_{\mathbf{zz}}$. The wavelet coefficients required in the approximate representation correspond to linear functions of the values $z(t)$. That is:

$$W_p\beta_{s,i,j} = W_pS\mathbf{z} \tag{136}$$

$$= L\mathbf{z} \tag{137}$$

where the matrix L is the product of the wavelet transform operator W_p and a "selection" matrix S which generates the vector $\beta_{s,i,j}$ from $\mathbf{z}$. Thus, to compute the correlation matrices required in the approximate representation, we need only compute functions of the form:

$$L_1R_{\mathbf{zz}}L_2^T = (L_1F^*)\Lambda(FL_2^T) \tag{138}$$

Indeed, as described in [43], the structure of the approximate representations and the stationarity of the GMRF allow us to compute the required correlations with only 2^p 2-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 18a illustrates the "wool" texture from [14]. Three sample paths of approximate representations of this field based on the Haar wavelet are shown in Figures 18b to 18d. The wool texture is corresponds to a fourth-order version of the model (132), with the coefficients given in Table 1.

Figures 18b and 18c correspond to zeroth and first-order approximations, respectively. Note in Figure 18c that some of the boundary effects apparent in Figure 18b have disappeared, due to the increase in the approximation order. Since the wool texture is actually Markov with respect to the fourth-order neighborhood of Figure 11, an exact representation of this field would require that boundaries of width equal to two be kept. To this end, the approximation shown in Figure 18d takes as state variables first-order approximations to these double width boundaries. Essentially all of the boundary effects present in the Figures 18b to 18c have been eliminated, and this representation appears to have retained

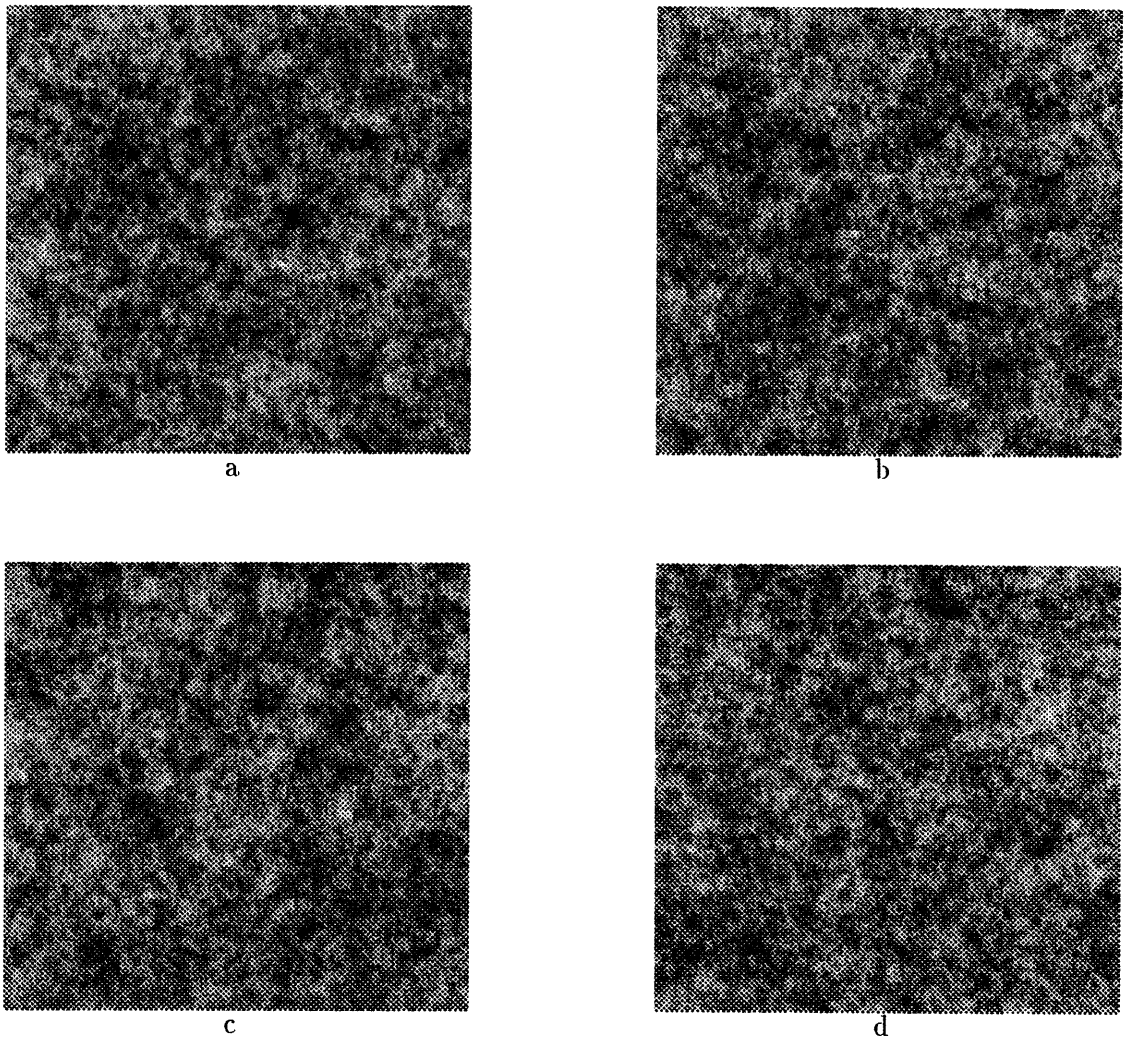


Figure 18: A sample path of a Gaussian MRF representing the "wool" texture of [14] is shown in (a). Figures 18b - 18d illustrate sample paths of approximate representations of this MRF based on the Haar wavelet. Zeroth and first-order approximations are used in (b) and (c),respectively. In (d), a first-order approximation based on boundaries of width two is used. Note that as the order of the approximate model is increased, the boundary effects disappear, and that for relatively low-order models an approximate representation which retains most of the qualitative and statistical features of the original MRF can be obtained.

(k,l)	$h_{k,l}$	(k,l)	$h_{k,l}$
(1,0)	0.4341	(0,2)	0.0592
(0,1)	0.2182	(-1,2)	-0.0302
(-1,1)	-0.0980	(1,2)	-0.0407
(1,1)	-0.0006	(-2,1)	0.0406
(2,0)	-0.0836	(2,1)	-0.0001

Table 1: Coefficients in the model (132) for the "wool" texture.

(k,l)	$h_{k,l}$	(k,l)	$h_{k,l}$
(1,0)	0.5508	(0,2)	0.0139
(0,1)	0.2498	(-1,2)	-0.0085
(-1,1)	-0.1164	(1,2)	-0.0058
(1,1)	-0.1405	(-2,1)	-0.0008
(2,0)	-0.0517	(2,1)	0.0091

Table 2: Coefficients in the model (132) for the "wood" texture.

the essential statistical and qualitative features of the exact representation used to generate Figure 18a.

Figure 19a illustrates the "wood" texture from [14], and three approximations of this MRF based on the Haar wavelet are shown in Figures 19b - 19d. The wood texture corresponds to a fourth-order version of the model (132), with the coefficients given in Table 2.

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 19b, the horizontal and vertical boundaries are represented with second and zeroth-order approximations respectively. In Figures 19c and 19d, the horizontal boundaries are represented with fourth and sixth-order approximations, respectively, whereas the vertical boundary is still 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 19c and 19d 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 (see Figure 11) and thus double width boundaries would be needed in an exact representation. The fields shown in Figures 19b to 19d, 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 19b to 19d. It was found, however, that there were no visual differences between the single and double width approximate representations.

Three approximations of the "wood" texture based on the Daubechies 8 wavelet de-

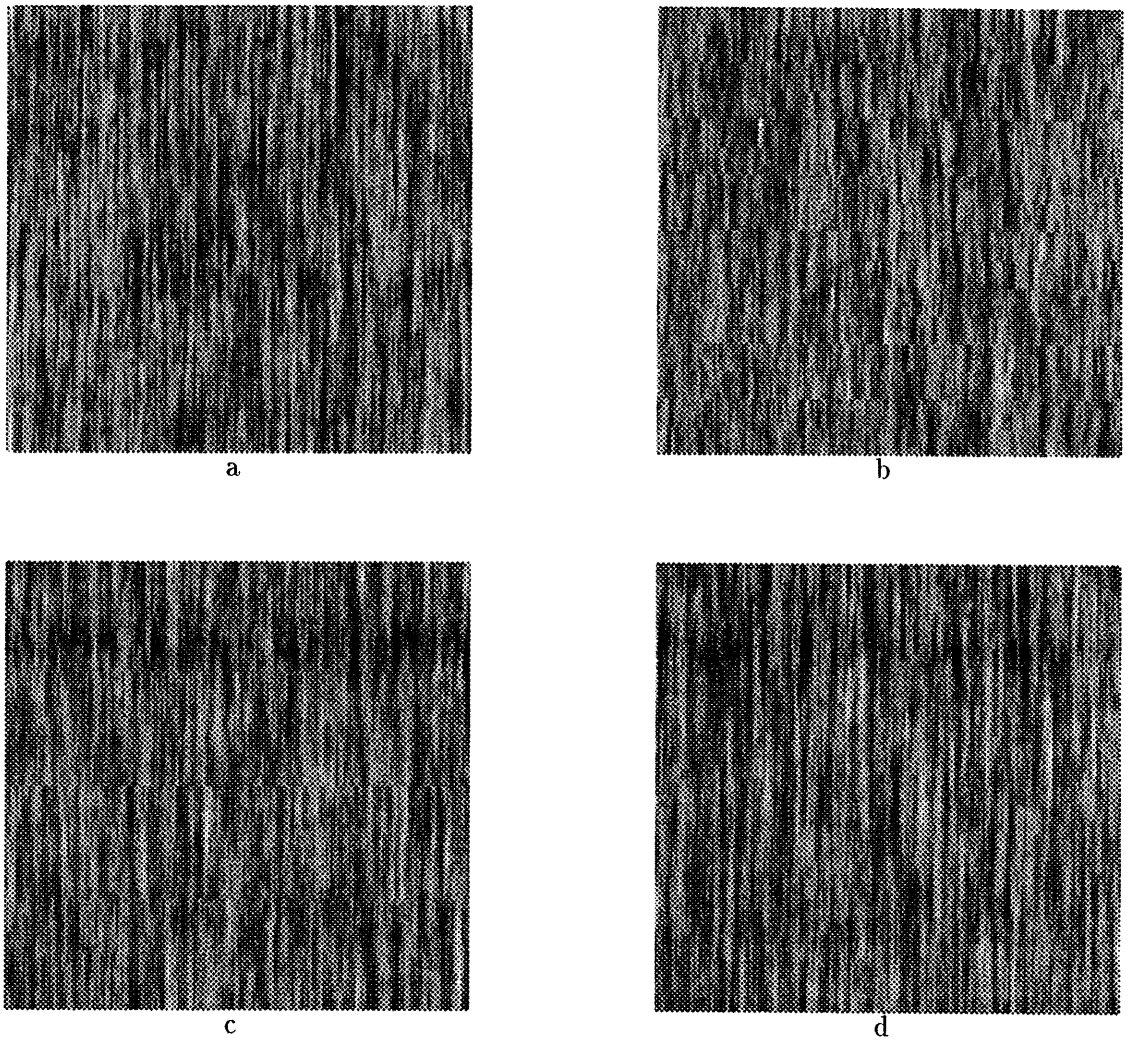


Figure 19: A sample path of a Gaussian MRF representing the "wood" texture of [14] is shown in (a). Figures 19b - 19d illustrate sample paths of approximate representations of the MRF based on the Haar wavelet. The structure of the MRF suggests using approximations which use relatively low order representations of vertical boundaries. The approximate representations used to generate Figures 19b - 19d used zeroth-order representations of the vertical boundaries, and second, fourth and sixth-order representations for the horizontal boundaries, respectively.

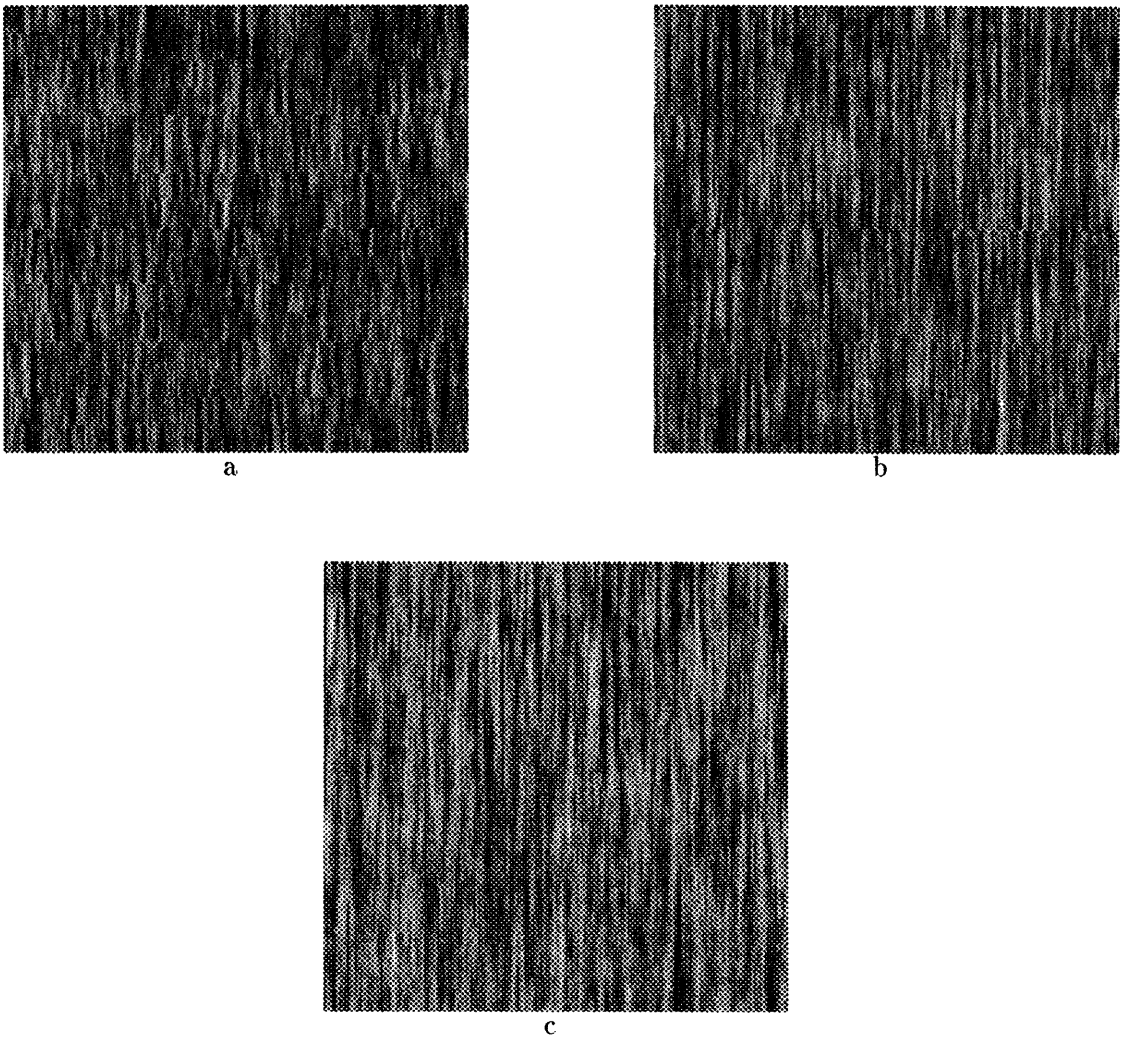


Figure 20: Figures 20a - 20c illustrate sample paths of approximate representations based on the Daubechies 8 wavelet, with the same orders of approximation as in Figures 19b - 19d.

scribed in [26] are illustrated in Figures 20a to 20c. The order of the approximations are identical to the orders for Haar approximations in the previous example. We note that there is no apparent difference between the approximations based on the Haar wavelet and the Daubechies 8 wavelet. 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 [16, 17, 18, 19]. In addition, as we briefly discuss in the next section, for GMRF's which have particular directions in which correlation structures are oscillatory rather than monotonically decaying (such as the one describing the "wood" texture), there

may be different choices of bases other than wavelets that lead to high fidelity models of even lower dimension.

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. Since for reasonable size fields this state dimension may be prohibitively large, we also introduced a class of multiscale *approximate* MRF representations based on 1-D wavelet transforms of the MRF along 1-D boundaries of multiresolution partitionings of the 2-D domain of interest. This family of models allows one to tradeoff complexity and accuracy of the representations, and provides a framework for the development of extremely efficient estimation and likelihood calculation algorithms. Examples demonstrated that for relatively low-order models, an approximate representation which retains most of the qualitative and statistical features of the original MRF can be obtained. Moreover, these approximate models lead to algorithms which have a complexity that is constant per pixel, in contrast to the algorithms normally associated with MRF's which have a per pixel complexity which increases with image size.

We feel that the results presented in the preceding section, together with the substantial flexibility of the multiscale modeling framework, both demonstrate the promise of this framework for image and multidimensional signal processing and also suggest a rich set of questions for further investigation. For example, our comparisons here between exact and approximate representations have focused only on their visual characteristics. It is important to develop other measures of the quality of these fields and in particular to define and analyze metrics that allow us to place the trade-off between fidelity and model order on a rational basis. For instance one natural basis for measuring the distance between two models corresponds to analyzing error probabilities associated with a binary hypothesis test in which one must decide whether a given random field is a sample path of the exact or approximate representation. In particular, if the probability of error is near to $1/2$, then the fields are nearly indistinguishable.

It is also important to investigate metrics directly related to performance in applications in which performance based on MRF models and on approximate models of varying orders can be compared. In particular, the GMRF texture models are often used in region labelling and image segmentation applications. It would be of interest to develop algorithms for these applications based on the approximate representations, and compare them in terms of performance and required computation to the standard algorithms. Note in particular, that while our models are approximations to MRF's, they may very well lead to algorithms, which are not only far simpler computationally but which also have equal or *better* performance. Specifically, an extremely important point is that all models, including MRF's and multiscale models, are *idealizations*, and thus it is not at all obvious that MRF based algorithms will perform better than those based on our models. Indeed, given the fact

that the multiscale modeling framework leads to efficient algorithms and is *richer* than the class of MRF's (in that any MRF can be approximated to any desired degree of accuracy using multiscale models), there is a strong argument in favor of the framework described here.

In addition, there are numerous other directions in which the ideas in this paper can be extended and further developed. For example there are strong motivations for the consideration of alternatives to wavelet transforms for the approximate representations used in our multiscale models. For instance, it is certainly possible to consider non-orthogonal multiresolution approximations to the values of MRF's along 1-D boundaries. Indeed, one possibility is to use linear or higher-order polynomial interpolation — i.e. to perform generalized mid-point deflection along each of the 1-D boundaries. Also, and perhaps more importantly, wavelet packet basis functions [62] may provide lower-dimensional approximations for some GMRF's, such as the "wood" texture discussed in the previous section, in which the random field has bandpass (i.e. oscillatory) rather than low-pass characteristics in one or more directions. Indeed, techniques such as in [25], suggest the idea of choosing the "optimal" set of basis functions within some class, where optimality might be measured in terms of the compactness of the representation, i.e. in terms of the order of the approximate model required to achieve an acceptable representation.

References

[1] K. Abend, T. Harley and L. Kanal, "Classification of Binary Random Patterns," IEEE Transactions on Information Theory, **11**:538-544, Oct. 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, Special issue on wavelet transforms and multiresolution signal analysis, **38**:766-784, April 1992.

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

[4] J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," J. 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] C. Bouman, "A Multiscale Image Model for Bayesian Image Segmentation," TR-EE 91-53, School of Electrical Engineering, Purdue University, December 1991.

[8] C. Bouman and M. Shapiro, "Multispectral Image Segmentation Using a Multiscale Model," Proc. ICASSP 92, Vol. 3, pp. 565-568.

[9] C. Bouman and M. Shapiro, "A Multiscale Random Field Model for Bayesian Image Segmentation," Submitted to IEEE Transactions on Image Processing, June 1992.

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

[11] P. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," IEEE Transactions on Comm., **31**:482-540, 1983.

[12] R. Chellappa and R. Kashyap, "Digital Image Restoration Using Spatial Interaction Models," IEEE Transactions On ASSP, **30**:461-472, June 1982.

[13] R. Chellappa And S. Chatterjee, "Classification Of Textures Using Gaussian Markov Random Fields," IEEE Transactions On ASSP, **33**:959-963, August 1985.

[14] R. Chellappa, B. Manjunath And T. Simchony, "Texture Segmentation with Neural Networks," in *Neural Networks for Signal Processing*, ed. B. Kosko, Prentice-Hall 1992, pp. 37 - 61.

[15] P. Chin Cheong and S. Morgera, "Iterative Methods for Restoring Noisy Images," IEEE Transactions on ASSP, **37**:580-585, April 1989.

[16] K. C. CHOU, A. S. WILLSKY, A. BENVENISTE AND M. BASSEVILLE, "Recursive and Iterative Estimation Algorithms for Multiresolution Stochastic Processes," Proc. of the IEEE CDC, Dec. 1989.

[17] K. C. CHOU, *A Stochastic Modeling Approach to Multiscale Signal Processing*, MIT Dept. of EECS Ph.D. Thesis, May 1991.

[18] K. C. CHOU, A. S. WILLSKY AND A. BENVENISTE, "Multiscale Recursive Estimation, Data Fusion and Regularization," MIT LIDS Report # LIDS-P-2085, December 1991. Submitted for publication to IEEE Transactions on Automatic Control.

[19] K. C. CHOU, A. S. WILLSKY AND R. NIKOUKHAH, "Multiscale Systems, Kalman Filters and Riccati Equations," submitted for publication to IEEE Transactions on Automatic Control.

[20] K. C. CHOU, S. GOLDEN AND A. S. WILLSKY, "Multiresolution Stochastic Models, Data Fusion, and Wavelet Transforms," MIT LIDS Report # LIDS-P-2110, May 1992. Submitted for publication to IEEE Transactions on Automatic Control.

[21] S. C. CLIPPINGDALE AND R. G. WILSON, "Least Squares Image Estimation on a Multiresolution Pyramid," Proceedings of the 1989 International Conference on Acoustics, Speech and Signal Processing.

[22] F. COHEN, Z. FAN AND M. PATEL, "Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Models," IEEE Transactions on PAMI, **13**:192-202, February 1991.

[23] F. COHEN, Z. FAN AND S. ATTALI, "Automated Inspection of Textile Fabrics Using Textural Models," IEEE Transactions on PAMI, **13**:803-808, August 1991.

[24] F. COHEN, "Modeling of Ultrasound Speckle with Application in Flaw Detection in Metals," IEEE Transactions on Signal Processing, **40**:624-632, March 1992.

[25] R. COIFMAN AND M. WICKERHAUSER, "Best-Adapted Wave Packet Bases," Numerical Algorithms Research Group, Dept. of Math., Yale University, 1990.

[26] I. DAUBECHIES, "Orthonormal Bases of Compactly Supported Wavelets," *Communications on Pure and Applied Mathematics*, **91**:909-996, 1988.

[27] M. H. A. DAVIS, *Linear Estimation and Stochastic Control.* John Wiley & Sons: New York 1977, p. 72.

[28] 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 PAMI, **6**:707-720, November 1984.

[29] H. DERIN AND P. KELLY, "Discrete Index Markov Type Random Processes," Proc. of the IEEE, **77**:1485-1509, October 1989.

[30] P. DOBRUSCHIN, "The Description of a Random Field by Means of Conditional Probabilities and Conditions of its Regularity," Theory of Probability and its Applications, **13**:197-224, 1968.

[31] P. DOERSCHUK, "Bayesian Signal Reconstruction, Markov Random Fields and X-Ray Crystallography," Center for Intelligent Control Systems Report # CICS-P-252, September 1990.

[32] P. FLANDRIN, "Wavelet Analysis and Synthesis of Fractional Brownian Motion," IEEE Transactions on Information Theory, Special issue on wavelet transforms and multiresolution signal analysis, **38**:910-917, March 1992.

[33] S. GEMAN AND D. GEMAN, "Stochastic Relaxation, Gibb's Distributions and the Bayesian Restoration of Images," IEEE Transactions on PAMI **6**:721-741, November 1984.

[34] A. K. JAIN, "Advances in Mathematical Models for Image Processing," Proc. IEEE, **69**:502-528, 1981.

[35] F. JENG AND J. WOODS, "On the Relationship of the Markov Mesh to the NSHP Markov Chain," Pattern Recognition Letters, **5**:273-279, July 1986.

[36] R. KASHYAP AND R. CHELLAPPA, "Estimation and Choice of Neighbors in Spatial Interaction Models of Images," IEEE Transactions on Information Theory, **29**:60-72, January 1983.

[37] S. LAKSHMANAN AND H. DERIN, "Simultaneous Parameter Estimation and Segmentation of Gibb's Random Fields using Simulated Annealing," IEEE Transactions on PAMI, **11**:799-813, August 1989.

[38] 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, Sept. 1990.

[39] B. C. LEVY, "Non-causal Estimation for Markov Random Fields," in Proc. International Symposium MTNS-89, Vol. 1: Realization and Modeling in System Theory, M. A. Kaachoek, J. H. van Schuppen, and A. Ran, eds., Birkhauser-Verlag, Basel, 1990.

[40] B. C. LEVY, "Regular and Reciprocal Multivariate Stationary Gaussian Reciprocal Processes Over $\mathcal{Z}$ Are Necessarily Markov," Univ. of California Davis, Dept. of Elect. Eng. and Comp. Sci. Technical report, March 1991.

[41] P. LÉVY, "Le Mouvement Brownien," Mem. Sc. Math., fasc. 126, pp. 1-81, 1954.

[42] M. LUETTGEN, W. KARL AND A. S. WILLSKY, "Optical Flow Computation via Multiscale Regularization," submitted for publication to IEEE Transactions on Image Processing, June 1992.

[43] M. LUETTGEN, "Applications of Multiscale Stochastic Models," MIT Dept. of EECS Ph.D. Thesis, in preparation.

[44] S. MALLAT, "A Theory for Multiresolution Signal Decomposition: the Wavelet Transform," IEEE Transactions on PAMI, **11**:674-693, 1989.

[45] H. S. MALVAR AND D. H. STAELIN, "The LOT: Transform Coding Without Blocking Effects," IEEE Transactions on ASSP, **37**:553-559, Apr. 1989.

[46] H. S. MALVAR, "Lapped Transforms for Efficient Transform/Subband Coding," IEEE Transactions on ASSP, **38**:969-978, June, 1990.

[47] B. S. MANJUNATH, T. SIMCHONY AND R. CHELLAPPA, "Stochastic and Deterministic Networks for Texture Segmentation," IEEE Transactions on ASSP, **38**:1039-1049, June 1990.

[48] B. S. MANJUNATH AND R. CHELLAPPA, "Unsupervised Texture Segmentation using Markov Random Field Models," IEEE Transactions on PAMI, **13**:478-482, May 1991.

[49] J. MARROQUIN, "Optimal Bayesian Estimators for Image Segmentation and Surface Reconstruction," M.I.T. Laboratory for Information and Decision Systems Report # LIDS-P-1456, May 1985.

[50] R. NIKOUKHAH, *A Deterministic and Stochastic Theory for Two-Point Boundary-Value Descriptor Systems.* MIT Dept. of EECS Ph.D. Thesis, Sept. 1988.

[51] L. ONSAGER, "Crystal Statistics 1: A Two Dimensional Model With an Order-Disorder Transition," Physical Review, Vol. 65, pp. 117 - 149, February 1944.

[52] THRASYVOULOS PAPPAS, "An Adaptive Clustering Algorithm for Image Segmentation," IEEE Transactions on Signal Processing, **40**:901-914, April 1992.

[53] B. POTAMIANOS AND J. GOUTSIAS, "Stochastic Simulation Techniques for Partition Function Approximation of Gibb's Random Field Images," Johns Hopkins University Dept. of Electrical and Computer Engineering technical report JHU/ECE 91-02, 1991. Also submitted to IEEE Transactions on Information Theory.

[54] Y. A ROSANOV, "On Gaussian Fields with Given Conditional Distributions," Theory of Probability and its Applications, **12**:381-391, 1967.

[55] M. J. T. SMITH AND T. P. BARNWELL, "Exact Reconstruction Techniques for Tree Structured Subband Coders," IEEE Transactions on ASSP, **34**:434-441, 1986.

[56] F. SPITZER, "Markov Random Fields and Gibbs Ensembles," American Mathematical Monthly, **78**:142-154, Feb. 1971.

[57] G. STRANG, "Wavelets and Dilation Equations: A Brief Introduction," SIAM Review, **31**:614-627, December 1989.

[58] D. TAYLOR, *Parallel Estimation on One and Two Dimensional Systems.* MIT Dept. of EECS Ph.D. Thesis, Feb. 1992.

[59] R. TENNEY AND A. WILLSKY, "Multi-Resolution Estimation for Image Processing and Fusion," AFIT Workshop on Wavelets and Signal Processing, TP-329, March 1992.

[60] D. TERZOPOULOS, "Image Analysis Using Multigrid Relaxation Methods," IEEE Transactions on PAMI, 8:129-139, 1986.

[61] A. H. TEWFIK AND M. KIM, "Correlation Structure of the Discrete Wavelet Coefficients of Fractional Brownian Motion," IEEE Transactions on Information Theory, **38**:904-910, March 1992.

[62] M. V. WICKERHAUSER, "Lectures on Wavelet Packet Algorithms," Technical report, Washington University Dept. of Mathematics, 1992.

[63] A. S. WILLSKY, "Opportunities and Challenges in Signal Processing and Analysis," in Proc. International Conference on Computer Science and Control, Paris, Dec. 8-11, 1992.

[64] J. WOODS, "Two Dimensional Discrete Markovian Fields," IEEE Transactions on Information Theory, **18**:232-240, March 1972.

[65] J. WOODS AND C. RADEWAN, "Kalman Filtering in Two Dimensions," IEEE Transactions on Information Theory, **23**:473-482, July 1977.

[66] G. WORNELL AND A. OPPENHEIM, "Estimation of Fractal Signals from Noisy Measurements," IEEE Transactions on Signal Processing, **40**:611-623, March 1992.