An Overlapping Tree Approach to Multiscale Stochastic Modeling and Estimation

William W. Irving*   Paul W. Fieguth†   Alan S. Willsky‡

Abstract

Recently a class of multiscale stochastic models has been introduced in which random processes and fields are described by scale-recursive dynamic trees. A major advantage of this framework is that it leads to an extremely efficient, statistically optimal algorithm for least-squares estimation. In certain applications, however, estimates based on the types of multiscale models previously proposed may not be adequate, as they have tended to exhibit a visually distracting blockiness. In this paper, we eliminate this blockiness by discarding the standard assumption that distinct nodes on a given level of the multiscale process correspond to disjoint portions of the image domain; instead, we allow a correspondence to overlapping portions of the image domain. We use these so-called overlapping-tree models for both modeling and estimation. In particular, we develop an efficient multiscale algorithm for generating sample paths of a random field whose second-order statistics match a prespecified covariance structure, to any desired degree of fidelity. Furthermore, we demonstrate that under easily satisfied conditions, we can "lift" a random field estimation problem to one defined on an overlapped tree, resulting in an estimation algorithm that is computationally efficient, directly produces estimation error covariances, and eliminates blockiness in the reconstructed imagery without any sacrifice in the resolution of fine-scale detail.

1   Introduction

Recently, a class of multiscale stochastic models has been introduced in which stochastic processes and fields are indexed by the nodes of a tree [2, 4]. These models provide a systematic way to describe random processes and fields that evolve in *scale*. The primary reason that this framework is useful is that it leads to extremely efficient, statistically optimal algorithms for signal and image processing. In particular, the statistical structure of these models leads directly to scale-recursive generalizations of both the Kalman filter and the Rauch-Tung-Striebel smoother [4]. The algorithm has demonstrated utility in confronting data assimilation problems of dauntingly large dimension; two applications where this approach has had considerable success include calculation of optical flow [15] and smoothing of ocean altimetric data [6]. In the latter work, for example, the authors were able to

estimate both ocean surface height *and* associated error statistics for a 512×512 grid, all in one minute on a SPARC-10.

In spite of the success of the multiscale approach to estimation with regard to computational efficiency, mean-square estimation error, and ability to supply error covariance information, the approach, as developed up to this point in time, has a characteristic that would appear to limit its utility in certain applications. Specifically, estimates based on the types of multiscale models previously proposed may exhibit a visually distracting blockiness [15]. The authors in [15] argue correctly that in many applications, the construction of fine-scale estimates is not supported by the quality of available data, and in such cases, only coarser scale estimates are statistically significant. In these cases, one should be suspicious of *any* fine-scale estimate, and any corresponding blockiness has a complete lack of statistical significance. However, in other applications, such as the problem of estimation of the ocean surface height [6] or the investigation of surface reconstruction in [8], there is an essential need for smooth estimates, so that surface gradients and normals can be calculated meaningfully.

Although estimate blockiness can be eliminated by simple post-processing (e.g., the application of a low pass filter, or the averaging of multiple, shifted multiscale-based estimates as in [7] or in a manner similar in spirit to the "cycle spinning" used in [5]), the resulting increase in smoothness comes at a price. In particular, the post-processing can render less clear the proper interpretation of error covariance information provided by the estimation algorithm, and it also limits the resolution of fine-scale detail in the post-processed estimate, since the added smoothness is achieved by spatial blurring. As an alternative, the work in [12, 17] has demonstrated that multiscale models can be constructed that produce arbitrarily accurate representations of broad classes of random fields, including those with considerable smoothness. However, to achieve a high level of smoothness, these methods require the use of multiscale processes of high dimension, thereby leading to a reduction in the significant computational advantages that the multiscale modeling framework offers.

Thus for applications in which the computational efficiency of the multiscale framework is desired, but where blockiness is unacceptable, there is a need for a new approach to both multiscale modeling and estimation. In this paper, we develop such an approach that

  (a) yields low-dimensional multiscale models that are quite faithful to prespecified random field covariance

structure to be realized, and thus admit an extremely efficient, optimal (or nearly optimal) estimation algorithm;

(b) retains one of the most important advantages of the multiscale estimation framework, namely the efficient computation of estimation error covariances;

(c) results in random fields and estimates without blocky artifacts;

(d) achieves these objectives without loss of resolution or fine-scale detail.

In contrast to the original multiscale processing [7, 15], which achieves objectives (a) and (b), and to standard multiscale processing with simple post-processing [15], which achieves objective (a) and partially achieves objective (c), our approach is superior in that it accomplishes all four objectives.

To describe our approach, we begin with a more careful look at the source of blockiness. Consider the standard quad-tree multiscale structure shown in Figure 1. Each level of this tree corresponds to a particular scale m, with larger m corresponding to finer resolution; the state $x(s)$ on scale $m(s)$ at any given node s represents an aggregate description of the subset of the finest-scale process that descends from the given node.[1] A critical property of multiscale models is that they are Markov: if $x(s)$ is the value of the state at node s, then conditioned on the value of $x(s)$, the sets of values of the states in the subtrees of nodes extending away from s are uncorrelated. This decorrelation leads both to efficient estimation algorithms and to the source of the problem with blockiness.

For example, consider the upper-left and upper-right quadrants of the image domain depicted in Figure 1. These two quadrants are separated at the root node at the coarsest level of the tree, and therefore all of the correlation between any two finer scale pixels in the two quadrants, such as s_1 and s_2 in Figure 2, must be completely captured in their common ancestor, namely the root node. The pixels s_1 and s_2 may be close physically, but they are separated considerably in terms of the distance along the tree to their nearest common ancestor node. High local correlation between such spatially close neighbors, as one might expect if the field being modeled has some

---

[1] We will use dyadic trees and quadtrees to illustrate our methods. All of these results generalize to q-th order trees and to trees having nonhomogeneous branching patterns.

level of regularity or smoothness, translates into the state at the root node having a high dimension, in essence to keep track of all of the correlations across quadrant boundaries.

One way to reduce dimensionality is to identify and retain only the principal sources of correlation across boundaries at each level on the tree. A procedure for doing this, developed in [12], allows us to build multiscale models of any desired fidelity. However, while this procedure by itself can yield low-dimensional models of sufficient fidelity for many applications, it cannot overcome the blockiness problem. In particular, neglecting even a small amount of correlation at a coarse level of the tree can cause noticeable irregularities across boundaries, and thus an additional element is required. In this paper, we introduce this new element by discarding the assumption that distinct nodes at a given level of a tree correspond to disjoint portions of the image domain and allowing the tree nodes to correspond to *overlapping* regions. As a consequence of this idea, which was first used in [6, 8], an image pixel at the finest scale may now correspond to several tree nodes at this finest scale. In this way we remove the hard boundaries between image-domain pixels, as now multiple tree nodes contribute to each of these pixels, reducing the tree distance between the nodes corresponding to these pixels and spreading the correlation that must be captured among a set of nodes. For obvious reasons, we refer to these as overlapped-tree models.

We use these overlapped-tree models for both modeling and estimation, as depicted in Figure 3. In both of these contexts, we start with assumed knowledge of the correlation structure P of some random field x.[2] Corresponding to this random field x, we devise a so-called *lifted-domain* version x_l, where this lifted-domain field lives at the finest-scale of an overlapped-tree multiscale representation of x. The mapping from x to x_l is denoted by $x_l = G_x x$, where G_x is one-to-many: the lifted-domain field x_l has more pixels than the image-domain field x. To map back from x_l to x, we devise an operator H_x so that the field $H_x x_l$ is guaranteed to have the desired level of smoothness. The correlation structure $P_l = G_x P G_x^T$ of the overlapped field x_l is approximately realized using the method developed in [12, 13]. Finally, we devise an operator G_y, analogous to G_x, that lifts the actual observations y to yield lifted-domain observations y_l of the random field x_l. These observations are then processed by the efficient multiscale tree algorithm to produce an estimate $\hat{x}_l$ which is then projected using H_x to yield $\hat{x}$, the desired estimate of the random field.

---

[2]For simplicity of notation we stack the random field such that x is a vector and P a covariance matrix.

In Section 2, we review the basic multiscale framework of [2, 4]. In Section 3 we introduce all of the components of our approach to modeling and estimation, and we characterize the optimality properties of the estimation procedure depicted in Figure 3. In Section 4 we describe an efficient implicit scheme for describing the projection operators to and from the overlapped domain, while in Section 5 we illustrate the effectiveness of our new approach to modeling and estimation by means of four examples, demonstrating not only that our method avoids blocky artifacts, but in fact does this without spatial blurring or compromising the advantages of multiscale models.

2   Introduction to the Multiscale Framework

2.1   Multiscale Models and Estimation

For a qth order tree (i.e., one in which each node has q offspring), we define both a fine-to-coarse shift operator $\bar{\gamma}$ such that $s\bar{\gamma}$ is the parent of node s, and a set of coarse-to-fine shift operators α_i, $i = 1, 2, ..., q$ such that the q offspring of node s are given by $s\alpha_1$, $s\alpha_2...$, $s\alpha_q$. Figure 1 depicts the relative locations of $s, s\bar{\gamma}$, and $s\alpha_1, s\alpha_2, s\alpha_3, s\alpha_4$ for a quadtree. The scale-recursive dynamics of interest are given by

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

where $x(s)$ is a vector-valued process on the tree and $w(s)$ represents white driving noise with identity covariance, independent of the initial condition $x(0)$ at the root node 0. We model $x(0)$ as a zero-mean random vector with covariance $P(0)$. If we interpret each level in the tree as a representation of one scale of the process, then we see that (1) describes the evolution of a process from coarse to fine scales.

In this paper we make use of the estimation algorithm [4, 15] which computes the linear least-squares estimate[3] $\hat{x}(s) \equiv E[x(s) \mid y(\sigma), \sigma \in \mathcal{M}]$, based on noisy observations

$$y(s) \;\;=\;\; C(s)x(s) \;+\; v(s), \tag{2}$$

where $C(s)$ is a matrix specifying the nature of the process observations, as a function of spatial location and

---

[3]If all of the random variables are jointly Gaussian, then $\hat{x}(s)$ is the conditional mean of $x(s)$ given $\{y(\sigma); \; \sigma \in \mathcal{M}\}$.

scale, and $v(s)$ represents additive white measurement noise. The algorithm also computes the associated error covariance $\tilde{P}(s)$. This algorithm takes explicit advantage of the Markovian structure of $x(s)$ on the tree and incorporates the measurements into the estimates via two recursive sweeps, with each sweep following the structure of the tree. Although the framework we describe applies to the general case, we focus here exclusively on the case of estimating a scalar random field (e.g., an image), given noisy (and possibly sparse) point measurements of the field. Specifically, we assume all attention focuses on the finest scale, so that observations are only available at that scale and only the fine-scale estimates are of interest. Furthermore, we assume that at this finest scale, both the state and the measurements are scalar valued.

To obtain insight into the efficiencies offered by multiscale models and the challenges we must meet, consider the complexity of the simulation and estimation of a multiscale process $x(s)$. There are three multiscale model parameters of interest in this discussion: (i) the number K of pixels in the image domain, (ii) the number N of finest-scale nodes in the multiscale model, and (iii) the maximal dimension n of any state vector $x(s)$ in the multiscale model. In previous applications, N has been identical to K; in the approach to be developed here, the overlapping nature of our trees leads to larger values of N, so that $K = rN$, where $0 < r < 1$ is a measure of the degree of overlap, with smaller r corresponding to more overlap and greater smoothness.

The two-sweep structure of our estimation algorithm implies that each node of the tree is visited exactly twice, where the computations at each node involve a number of floating point operations proportional to the cube of the state dimension. Thus, application of the estimation algorithm requires a total of $\mathcal{O}(n^3 N)$ floating point operations. Similarly, the simulation of the coarse-to-fine recursion in (1) requires a total of $\mathcal{O}(n^2 N)$ floating point operations.[4] These complexity figures imply that a serial implementation requires a total computational time per image pixel of $\mathcal{O}(n^3/r)$ for estimation and $(\mathcal{O}(n^2/r))$ for simulation. The point here is that we can achieve dramatic computational benefit as long as the maximal dimension n of the state model and the amount of overlap (as measured by $1/r$) are not too large. As we will illustrate, the procedure we describe here allows us to meet these criteria.

---

[4] The fact that estimation is $\mathcal{O}(n^3)$ while simulation is only $\mathcal{O}(n^2)$ arises because the former involves matrix products, while the latter involves only matrix-vector products.

2.2   Realization of Multiscale Models

Our approach to building overlapped models makes use of a technique described in [12,13] for taking a specified covariance structure for a random field and constructing a multiscale model so that the set of values at the finest-scale nodes have statistics that approximately match the specified covariance structure. The problem of constructing such a model is the multiscale generalization of the problem of stochastic realization for time series, and the technique developed in [13], is based on the statistical concept of canonical correlations used in building time-series models [1]. As discussed in [1,13], the key to constructing a recursive model for a time series $z(t)$ is the specification of the state $x(t)$ at each time t. If $z_p(t)$ denotes the past of the process at time t and $z_f(t)$ the future then the components of $x(t)$ represent a set of linear functionals of the past, so that conditioned on $x(t)$, $z_p(t)$ and $z_f(t)$ are uncorrelated. Of course, the dimension of the needed state is closely tied to the correlation structure of the process, and for many applications, one can expect that an exact realization of the specified $R_{zz}(l)$ will require an unduly high state dimension. Thus, in addition to a method for finding and characterizing the state in an exact realization, there is also a need for a way in which to measure the relative importance of the components of the state, so that a decision can be made about which components to discard in a reduced-order realization.

For time series, canonical correlations applied to the covariance P_z of $z = \left(z_p(t)^T \quad z_f(t)^T \right)^T$ deals simultaneously with both of these issues. Specifically, through both a normalization and an orthogonal transformation, P_z is transformed to the form

$$\begin{pmatrix} I & D \\ D & I \end{pmatrix},$$

where I is the identity matrix and $D = \mathrm{diag}(\rho_1, \rho_2, \ldots, \rho_n)$ is the diagonal matrix of canonical correlations. In this form, which can be determined more directly from an SVD of a normalized version of the cross-correlation between $z_p(t)$ and $z_f(t)$ [13], we can identify the most highly correlated components of the past and future (corresponding to the largest ρ_i). By retaining all the linear combinations of the past corresponding to nonzero ρ_i, or only the largest of the ρ_i, we can construct the exact state $x(t)$ or an approximate state of any desired dimension.

Our multiscale context requires a significant generalization of these ideas. For example, the state at the node s in the figure must decorrelate not just two sets of random variables, but 5 sets of the process values, one for each of the sets of nodes connected to s through its children and one set corresponding to the nodes connected to s through its parent. This generalization is developed in [13], in which it is also shown how, once $x(s)$ has been defined at each node, the parameters of the model in (1), i.e., $A(s)$, $B(s)$ and $P(0)$ can be computed.

3 Modeling and Estimation with Overlapping Trees

In this section we identify the operators required in our approach to multiscale modeling and estimation with overlapping trees, and in particular describe their properties and interrelationships. We will also prove that *any* suboptimality in our approach to estimation can be completely traced to our use of an approximate model to realize the correlation structure of the overlapped field x_l. That is, if an exact realization is used in the overlapped domain, the overall procedure depicted in the bottom half of Figure 3 yields the optimal estimates.

3.1 Modeling of Random Fields with Overlapped Tree Processes

Consider the problem of simulating a zero-mean random field x with covariance P. From a computational point of view, this simulation problem poses nontrivial challenges and has been the focus of considerable research. One notable case in which efficient techniques do exist is for the simulation of stationary random fields defined on regularly sampled toroidal lattices, since in this case, the 2-D FFT can be used. However, for most other types of fields, simulation can be quite complex. For example, an approach based on computing $x = P^{1/2}w$, where $P^{1/2}$ is the square root of the covariance and w is a random vector having identity covariance requires computing the matrix square root $P^{1/2}$, which has complexity $\mathcal{O}(K^3)$ for a random field of K points. In contrast, as discussed in Section 2.1, the simulation of a random field having a multiscale model is extremely fast.

Our construction of a simulation procedure involves two distinct steps. In the first step, we specify the matrix G_x, which serves to lift the random field x into another random field x_l via

$$x_l \quad = \quad G_x x. \tag{3}$$

which acts as a redundant representation of x, having more pixels than the original field. The matrix G_x has a considerable amount of sparse structure, as we discuss in Section 4; G_x also has a left inverse H_x satisfying certain smoothness properties to be discussed shortly. In the second step, we use the method in [13] to build a low-dimensional multiscale model whose finest-scale statistics are an accurate approximation to the statistics of x_l. From (3), we see that the covariance of x_l is

$$P_l \;=\; G_x P G_x^T. \tag{4}$$

The covariance Σ_l of ξ_l, the random field living at the finest scale of the multiscale model that we construct, is approximately equal to P_l, the degree of approximation controlled by the procedure in [13]. Finally, to generate a sample function of a random field ξ having approximately the same statistics as x and with the desired smoothness built into H_x, we generate ξ_l using the efficient simulation procedure for processes on trees and then apply the operator H_x:

$$\xi \;=\; H_x \xi_l. \tag{5}$$

The problem then is to specify G_x and H_x, so that (i) they are sparse and local, (ii) H_x achieves the desired smoothness without spatial blurring, (iii) the resulting multiscale model is of sufficiently low dimension that simulation can be done efficiently, and (iv) the statistical approximation is sufficiently accurate so as to lead to sample functions with the desired characteristics.

To illustrate these ideas, consider a very simple 1-D example of a random process of length 3. Let $x^T = [x_1, x_2, x_3]^T$, with

$$E\left[xx^T\right] = P \equiv \begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix} \tag{6}$$

Suppose that we wish to develop an overlapped model for x, indexed on a dyadic tree having four finest-scale nodes. On the right of Figure 4 we depict such a tree with an indication of the subsets of real, physical points (i.e., subsets of $\{1, 2, 3\}$) to which each node corresponds. Thus, the top node corresponds to all three points (i.e., $\{1, 2, 3\}$) and the two nodes at the second level correspond to $\{1, 2\}$ and $\{2, 3\}$ respectively. At the bottom level there is a single node corresponding to signal point 1 and another for 3, but there are *two* nodes corresponding to

2. That is, in the lifted domain on the tree, signal point 2 is lifted to two finest-scale tree nodes. Thus if we order the four fine-scale nodes from left to right, and we view our lifting process as simply copying the value of signal point 2 to both of the tree nodes to which it corresponds, we are led to define

$$G_x \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7}$$

This example illustrates the constraints that we place on any lifting matrix G_x: it consists entirely of zeros and ones, each column has at least one nonzero entry, and each row has exactly one nonzero entry. These conditions ensure that every pixel in the original domain corresponds to at least one finest-scale node in the overlapped domain, and that every finest-scale node in the overlapped domain corresponds to exactly one pixel in the original domain. This lifting process can be associated naturally with the overlapping structure as illustrated in Figure 4. Depending on how one chooses an overlapping structure, a different lifting operator will generally result. In Section 4 we present an implicit method for the specification of G_x given a desired overlapping structure.

The fact that $H_x G_x = I$ and our imposed constraints on G_x, lead to an important constraint on the structure of H_x, namely that the value at any given point in the original-domain is equal to a convex combination of the values of the finest-scale nodes corresponding to that point. For example, with G_x as in (7), the possible choices for H_x are of the form

$$H_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a & b & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

where $a + b = 1$. Here a and b are weights placed on the values at the two nodes corresponding to signal point 2 in order to specify x_2. For example, equal weighting $a = b = 1/2$ would intuitively lead to the most smoothness in the correlation structure from x_1 through x_3. It is important to emphasize that the averaging implied by (8) is not at all the same as spatial averaging, since we average only those tree points corresponding to the *same* point in real space. In Section 4 we will also describe how H_x can be constructed directly from a specified overlap structure.

10

3.2 Estimation of Random Fields with Overlapped Tree Processes

Suppose that we wish to estimate a zero-mean random field x with covariance P based on

$$y = Cx + v \tag{9}$$

where the components of the measurement noise vector v are uncorrelated so that its covariance R is diagonal, and each component of y represents a measurement of an individual pixel so that each row of C has exactly one nonzero entry. Without loss of generality, we also assume that any pixel has at most one measurement associated with it.[5] This is equivalent to assuming that each column of C has at most one nonzero entry, so that C is a so-called (weighted) selection matrix and has full row rank.

From basic results in estimation theory we know that if x is zero-mean with covariance P, then

$$\hat{x} = PC^T \left(CPC^T + R \right)^{-1} y \equiv Ly \tag{10}$$

and the resulting error variance is given by

$$\tilde{P} = P - PC^T \left(CPC^T + R \right)^{-1} CP = P - LCP \tag{11}$$

For a K-pixel field, the explicit calculation of either L or $\tilde{P}$ is generally $\mathcal{O}(K^3)$ and the calculation of $\hat{x}$ is $\mathcal{O}(KM)$ where M is the number of measurements. Virtually the only case in which this computational load can be reduced to a practical level is when the field x is stationary, and we have dense, regularly sampled measurements of identical quality (implying that C and R are both multiples of the identity); in this special case FFT methods reduce the load to $\mathcal{O}(K log K)$. However in other cases, the $\mathcal{O}(K^3)$ computational load for the explicit calculation of L cannot be reduced, and the usual approach is to turn to iterative methods for the computation of $\hat{x}$. While iterative methods can significantly reduce the computational load of calculating $\hat{x}$, the calculation of error covariance information is computationally prohibitive. In contrast, the multiscale estimation algorithm described in Section 2.1 has a computational load of $\mathcal{O}(K)$ to compute both $\hat{x}$ and the diagonal elements of $\tilde{P}$.[6]

In addition to specifying G_x and H_x and a multiscale model for ξ_l, the approach illustrated in the bottom half

---

[5] If there are repeated measurements of a single pixel we can replace them by a single aggregate measurement obtained by taking a weighted average of the data.

[6] Other elements of $\tilde{P}$ may be computed using the results of [16].

of Figure 3 requires a lifting operator G_y for the measurements:

$$y_l = G_y y \tag{12}$$

and a lifted measurement model

$$y_l = C_l x_l + v_l. \tag{13}$$

Moreover, for the multiscale estimation algorithm to be applicable to estimating x_l based on y_l, (13) must represent uncorrelated observations of individual fine-scale tree nodes. That is, each row of C_l must have only one non-zero entry and the covariance R_l of v_l must be diagonal.

Since G_x associates each pixel with a set of fine-scale nodes, a natural choice for C_l is specified by requiring that if a real measurement is made at a particular pixel, then lifted measurements should be specified at each of the fine-scale tree nodes corresponding to that pixel. For example, for the three-point process illustrated in Figure 4, suppose that we have measurements of x_1 and x_2, namely

$$y \equiv \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad C \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \quad R \equiv \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix} \tag{14}$$

Then, in our lifted domain we should have *three* measurements, one corresponding to the single node associated with x_1, and *two* corresponding to the nodes associated with x_2. That is,[7]

$$C_l \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{15}$$

An obvious question at this point is how to create three measurement values on the tree when only two real measurements are available. The answer here is that we simply *copy* the actual measurement value at any pixel to all fine-scale nodes associated with that pixel. In our example,

$$G_y \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad y_l = G_y y \equiv \begin{bmatrix} y_1 \\ y_2 \\ y_2 \\ 0 \end{bmatrix} \tag{16}$$

---

[7]We shall find that having a measurement at *every* finest-scale node will make the precise description of operator G_y much simpler notationally. Consequently the measurement matrix C_l is padded with zero-rows (i.e., dummy measurements) to make it diagonal. It must be stressed that this is purely a notational matter and has no consequences on the theory or practical implementation of overlapping tree algorithms.

At first glance, this procedure appears to create a problem: for the multiscale estimation algorithm to work, the measurements at distinct nodes must have uncorrelated errors. With y_l and C_l defined as in (16) and (15) this certainly does not hold, since two of the "measurements" are identical. Nevertheless, we simply *model* these two measurements as being distinct, each of the state at the corresponding node, with uncorrelated measurement errors. However, this appears to create another difficulty. Specifically, by modeling y_l in this way we appear to be saying that we have more information than we actually do; in our example we now have two measurements of the nodes corresponding to x_2. To compensate for this, we need to ensure that the total information in these two measurements is the same as in the single real measurement. We accomplish this simply by doubling the corresponding measurement noise variances in our model for each of the replicated measurements; specifically, given R in (14) we define

$$R_l \equiv \begin{bmatrix} 1 \cdot 3 & 0 & 0 & 0 \\ 0 & 2 \cdot 4 & 0 & 0 \\ 0 & 0 & 2 \cdot 4 & 0 \\ 0 & 0 & 0 & * \end{bmatrix} \tag{17}$$

(where $*$ corresponds to the dummy measurement of (15); the value of $*$ is irrelevant).

The general case proceeds exactly as in this example. For each real pixel measurement we have an analogous measurement for *each* of the tree nodes corresponding to that real pixel. Thus if the jth component of y is $y_j = \alpha_j x_i + \text{noise}$ (where x_i is a component of x) then y_l will have measurements of the form $(y_l)_n = \alpha_j (x_l)_n + \text{noise}$ for each n such that finest-scale node $(x_l)_n$ corresponds to the real pixel x_i. Furthermore, for convenience, for any real pixels x_i that are not measured, we include dummy measurements for each node $(x_l)_n$ corresponding to x_i, but with $\alpha_j = 0$. Since C is a weighted selection matrix, so is C_l. To provide a formula for G_y, note that there are apparently two distinct ways in which x affects y_l: (i) through $x_l = H_x x$ and $y_l = C_x x_l + v_l$, and (ii) through $y = Cx + v$ and $y_l = G_y y$. A logical requirement then is that both effects of x on y_l should be the same, i.e., that $C_l G_x = G_y C$. Since C has full row rank, it follows that

$$G_y = C_l G_x C^T (CC^T)^{-1}. \tag{18}$$

While this expression for G_y is correct, its simple structure is obscured. However, noting that $(CC^T)^{-1}$ is diagonal and that the weights in C_l are the same as those in C, it follows that G_y is a lifting matrix, and in fact

consists of the subset of the columns of G_x corresponding to pixels at which we have measurements. This is consistent with our example: we assign values to the lifted measurements simply by replicating the appropriate original-domain measurement values.

To specify R_l, let $g(j)$ denote number of ones in the jth column of G_y; i.e., the number of times that the jth original-domain measurement is replicated in the overlapped domain. We then define R_l to be a diagonal matrix whose ith diagonal entry is given by

$$R_l(i,i) = g(j)R(j,j),$$

where j is the unique index for which $G_y(i,j) = 1$, (i.e., fine scale node i corresponds to pixel j). This choice for R_l, which is exactly what was done in our simple example, provides the observation covariance amplification required in the lifted domain to offset the apparent increase in information caused by the replication of measurement. This is stated more precisely in the following identity:

$$G_y^T R_l^{-1} G_y = R^{-1} \tag{19}$$

Proposition 1 *Let x be a random field with covariance P and let $y = Cx + v$ be a set of measurements with C a weighted selection matrix and R, the covariance of v, diagonal. Suppose we then choose G_x, H_x, G_y, C_l and R_l as just described. Then the optimal estimate $\hat{x}$ of x based on y can either be computed directly or by lifting, performing optimal estimation in the lifted domain, and then projecting. That is, if $\hat{x} = Ly$, and $\hat{x}_l = L_l y_l$, then*

$$PC^T \left(CPC^T + R\right)^{-1} = L = H_x L_l G_y = H_x P_l C_l^T \left(C_l P_l C_l^T + R_l\right)^{-1} G_y \tag{20}$$

where P_l is defined in (4). Moreover, if $\tilde{P}$ denotes the estimation error covariance in estimating x based on y, and $\tilde{P}_l$ the estimation error covariance in estimating x_l based on y_l, then

$$\tilde{P} = H_x \tilde{P}_l H_x^T \tag{21}$$

The proof is given in Appendix A. This proposition states that if we perform optimal estimation in the lifted domain using the correct covariance P_l for x_l, then the overall procedure depicted in the lower half of Figure 3 yields the optimal estimate for x based on y. Thus, any suboptimality in our actual implementation is completely traceable to approximations in building a low-order model for x_l.

14

4 Specification of the Overlapping Framework

The key quantities in building an overlapping framework are G_x and H_x: once these are determined P_l is fixed, so that a realization may be constructed, and C_l, G_y, and R_l can be computed as we have described. In this section we describe a flexible, implicit and efficient method for specifying G_x and H_x. For simplicity in exposition and notation, we focus on a basic case that conveys the main ideas, namely the representation of 1-D random processes with dyadic overlapping tree models having a spatially uniform overlap structure. That is, for any two nodes s_1 and s_2 on the same scale of the tree, the manner in which their descendants overlap must be the same. For a model having this structure and also having $M + 1$ scales, G_x and H_x can be specified *completely* in terms of only M parameters.

Recall that each node on the multiscale tree is associated with a connected interval of points in the original domain. We denote the width of this interval, for a node at scale m, by w_m. This is illustrated in Figure 5 which also depicts the geometry of the overlap of the intervals associated with the two children of any given node. We denote the amount of this overlap between sibling nodes at scale m by $o_m \geq 0$, and we require that sibling nodes do not completely overlap:

$$0 \leq o_m < w_m, \qquad m = 1, 2, \ldots, M. \tag{22}$$

From Figure 5 we see that w_m and o_m are related by the following recursion:

$$w_{m-1} = 2w_m - o_m. \tag{23}$$

The M overlap parameters $\mathcal{O} \equiv \{o_1, o_2, \ldots, o_M\}$ provide a complete characterization of the overlap structure of the tree. However the values of M and $\mathcal{O}$ are not unconstrained. Clearly, the length K of the 1-D sequence being represented imposes the constraint that

$$M \geq \lceil \log_2 K \rceil, \tag{24}$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to x. For any fixed M satisfying (24), the overlap parameters $\mathcal{O}$ are implicitly constrained by two boundary conditions on the recursion (23). First, each node on the finest level of the tree must correspond to a single pixel:

$$w_M = 1. \tag{25}$$

Second, the root node of the multiscale tree must be associated with the entire random field:

$$w_0 = K. \tag{26}$$

The constraints (22)–(26) still leave some degrees of freedom in specifying $\mathcal{O}$. In our examples in Section 5, we eliminate these by additionally constraining the so-called fractional overlap, o_m/w_m to be approximately constant as a function of scale.[8]

With regard to selecting a value for M, note that as the value M is increased, for a fixed value of K, the amount of overlap at each scale must also increase, in order to fulfill the boundary conditions (25) and (26). Since a greater amount of overlap leads to greater smoothness, increasing M leads to smoother realizations of a given field. However, as M increases, the complexity of carrying out simulation and estimation also increases. Thus, there is tradeoff involved in choosing a value for M that is typically best resolved by a combination of engineering judgment and numerical experimentation.

The matrix G_x follows uniquely once values for M and $\mathcal{O}$ have been chosen. Specifically, thanks to the constraints on G_x, we know that its k-th row will have a single non-zero entry having a value of one. If we let s_k denote the k-th node at the finest scale of our overlapping tree, then this node will correspond to some index l_k in the 1-D process being represented, and so

$$G_x(k,l) = \begin{cases} 1 & l = l_k \\ 0 & \text{otherwise} \end{cases}$$

The index l_k can be determined directly from M and $\mathcal{O}$. Clearly, there is a unique path from the root node 0 to the node s_k, where this path can be described as a sequence of M downward-shift operations:

$$s_k = 0\alpha_{j_1}\alpha_{j_2}\ldots\alpha_{j_M} \qquad j_m \in \{1, 2\}. \tag{27}$$

Here, $\sigma\alpha_1$ and $\sigma\alpha_2$ represent the left and right children, respectively, of node σ, and

$$k = \sum_{m=1}^{M}(j_m - 1)2^{M-m}. \tag{28}$$

---

[8]The fractional overlap cannot generally be made exactly constant, since the parameters w_m and o_m must take on integer values.

Finally, from our earlier discussion of overlap geometry a direct calculation shows that

$$l_k = \sum_{m=1}^{M} (j_m - 1)(w_m - o_m). \tag{29}$$

Applying the same procedure to each row of the matrix G_x yields the entire matrix.

The construction of H_x, while constrained by the choice for $M, \mathcal{O}$ and the fact that $H_x G_x = I$, still has degrees of freedom to be specified. To enforce our restriction that H_x perform no spatial averaging we require that nodes at the finest scale of the tree be mapped only to the pixels to which they correspond. Thus if $G_x(i, j) = 0$, then we require that $H_x(j, i) = 0$. One way to meet this constraint is to let H_x be the Moore-Penrose pseudo-inverse of G_x. However, it is possible to devise a matrix H_x that actually does a better job of smoothing.

To describe the H_x that we use, consider the two child nodes shown in Figure 6(b) and a pixel that lies within the overlapping regions of these two nodes (e.g., the pixel marked $\star$ in the figure). We need to specify the contributions of the two child nodes (and their descendants) in determining the value of pixel $\star$; for example, as indicated in the figure, the left child is given a weight of $\frac{1}{4}$ and the right child a weight of $\frac{3}{4}$. Thus the right child (and its descendants) will have a contribution three times that of the left child to the value at $\star$. In order to maintain a total contribution of unity at each pixel, we will normalize the contributions at each pixel to sum to one; these normalized values will be referred to as *relative* contributions. We achieve smoothness in H_x by tapering the relative contributions of a node towards zero as one approaches an overlapped end of the interval associated with the node; one such tapering is sketched in Figure 6(a).

Suppose that the procedure outlined in the preceding paragraph has been applied to all nodes on all scales. To illustrate how H_x is determined from these contributions, consider a node s_k on the finest scale and define k and l_k as in (28),(29). The participation of node s_k on the finest scale is determined as the product of all relative contributions associated with all ancestors of s_k. This construction is illustrated in Figure 7 for an overlapping tree representation of a 1-D process having four points: (a, b, c, d). Consider finest scale node $s = \circledb$ (second from the left end of the tree). The participation of s in determining the value at point b is given by the product of the numerical values above each $\circledb$ in Figure 7. Thus the participation of s is equal to $1 \cdot \frac{1}{2} \cdot \frac{2}{3} \cdot 1 = \frac{1}{3}$; so the weight in H_x associated with s is $\frac{1}{3}$. The weights in H_x corresponding to each of the finest-scale nodes are

17

shown in Figure 7.

For all but the smallest estimation problems, a dense representation of the G_x and H_x matrices is impractical. In fact, for large multidimensional problems even a sparse representation, using the fact that each row of G_x and column of H_x contain only one nonzero entry, may be too large. However, the M parameters in the parameterization $\{M, \mathcal{O}\}$ form an implicit representation of G_x and H_x. We have found the on-line construction of G_x and H_x from $\mathcal{O}$ to be so rapid that we have exclusively used this latter representation in our software.

There are two possible extensions of this overlap specification which allow greater flexibility in representing 1-D processes. First, rather than using dyadic trees we can also use q-adic trees. As described in [9] the implicit procedure for specifying G_x and H_x from M and $\mathcal{O}$ can be directly extended to this case. In addition, it is possible to consider nonuniform overlap structures, i.e., structures in which some regions have more overlap than others. In this case the specification of the structure requires more than a single, common overlap o_m at each scale, and as a result the specification of G_x and H_x, while possible, is more complex. We have not found this added complexity warranted in any application.

Finally, there is the extension to the representation of 2-D random fields. By considering each of the dimensions separately, the procedure we have described can be directly extended. Specifically, instead of using a dyadic tree, we use a quadtree, with a splitting of regions occurring in each of two dimensions. In this case it is certainly possible to use a separate set of M overlap parameters $\mathcal{O}^1$ and $\mathcal{O}^2$ for each dimension. However in all applications we have considered we have found the use of a single set of overlap parameters for both dimensions to be adequate. In this case a straightforward extension [9] to the bookkeeping described for the 1-D case allows us to specify G_x and H_x implicitly, and it is this implicit specification that is used exclusively for the results presented next.

5   Experimental Results

In this section we demonstrate four applications of our overlapping tree framework. The basis for all of these examples is a particular Markov random field (MRF) model chosen because its strong anisotropy presents a most severe challenge in overcoming blockiness. The statistics of the MRF $z(i, j)$ of interest are implicitly governed

k =	-2	-1	0	1	2
2		-0.0085	0.0139	-0.0058	
1	-0.0008	-0.1164	0.2498	-0.1405	0.0091
l = 0	-0.0517	0.5508		0.5508	-0.0517
-1	0.0091	-0.1405	0.2498	-0.1164	-0.0008
-2		-0.0058	0.0139	-0.0085	

Table 1: Coefficients $\{h_{k,l}\}$ of the Markov random field "wood" model [14].

by the following autoregressive model:

$$z(i,j) = \sum_{k,l \in \mathcal{D}} h_{k,l} z(i-k, j-l) + v(i,j). \tag{30}$$

In this equation, $v(i,j)$ is a Gaussian noise process having the following correlation structure:

$$E\left[v(x,y)v(x+k, y+l)\right] = \begin{cases} \sigma^2 & k = l = 0 \\ -\sigma^2 h_{k,l} & (k,l) \in \mathcal{D} \\ 0 & (k,l) \notin \mathcal{D} \end{cases} \tag{31}$$

$$E\left[v(x,y)z(x+k, y+l)\right] = \begin{cases} \sigma^2 & k = l = 0 \\ 0 & \text{Otherwise} \end{cases} \tag{32}$$

where the field z is normalized to have unity variance, and where $\mathcal{D}$ denotes the set of offsets of the neighbors of any given field point (i,j). The specific choice of coefficients $\{h_{k,l}\}$ to be used in our examples are those of the "wood" texture [14], tabulated in Table 1.

In Figure 8(a), we display a 64 by 64 pixel sample function based on the "wood" texture coefficients and assuming that the MRF lies on a toroidal lattice so that FFT techniques can be employed. The image possesses an obvious grain – that is, a much stronger correlation in the vertical direction than in the horizontal. The long vertical correlation length is of particular interest: it is such correlations which non-overlapped multiscale trees find difficult to preserve, even using relatively high order models [17].

The overlapping-tree construction outlined in the previous sections requires three quantities to be specified. First, we must specify the order, q, of the tree (here we use a quadtree, $q = 4$). Secondly, we must specify the number of scales M in the tree. For a 64 by 64 pixel field, *non*-overlapped trees have $M = 7$ scales. In our examples with overlapped trees we will set $M = 8$ scales and use an overlap parameterization $\mathcal{O} = \{10, 5, 3, 2, 1, 0, 0\}$, which is consistent with (25) and (26), and renders approximately constant the fractional overlap o_m/w_m. Finally, we must specify the order k of the multiscale model to be constructed using the method of [13].

5.1 Modeling Example

While we expect that the principal use of our multiscale models will be in estimation and statistical analysis, we begin with an example of simulating random fields, illustrating the improvement using overlapped models. We have constructed approximate multiscale models for the wood texture MRF using two different choices for the pair (M, k), such that the computational efforts required to simulate the two fields in Figures 8(b),(c) are the same. The first of these has $M = 7$ and $k = 64$; the resulting blockiness in Figure 8(b) is clear. The second realization, in Figure 8(c), is based on $M = 8$ and $k = 16$; this model is overlapping, and we can see that there are no blocky artifacts.

5.2 Estimation: Densely Sampled Field, Homogeneous Model

Consider the case in which we have dense, regularly sampled, equal quality measurements on a toroidal lattice so that the exact optimal estimate can be calculated using FFTs. The original texture shown in Figure 8(a) was corrupted to 0dB SNR by white Gaussian noise, and estimated in three different ways: (i) using an optimal FFT technique (Figure 9(a)), (ii) using a non-overlapped multiscale tree ($M = 7$) with a multiscale model of order $k = 40$ (Figure 9(b)), and (iii) using an overlapped tree ($M = 8$) and a multiscale model of order $k = 16$ (Figure 9(c)). The model orders of the two multiscale techniques were chosen so that the computational burden for estimation is the same for both. One measure of estimator performance is the degree to which the MSE is reduced from the original noisy image, relative to the reduction provided by the optimal least-squares estimator. The non-overlapped estimator and the overlapped estimator respectively reduce the MSE by 98.6% and 98.0% of the optimal MSE reduction. On the other hand, a visual comparison between Figures 9(b) and (c) shows clearly the presence of blocky artifacts for the non-overlapped case but no such artifacts for the overlapped model. Thus the overlapped model is decidedly superior if the elimination of such artifacts is an important concern.

Although the FFT technique is both efficient and optimal in terms of MSE, it suffers from a limited applicability to special circumstances. In particular, the examples are presented in the following two subsections, irregularly sampled measurements and a spatially varying prior model, preclude the use of the FFT but may be solved using our multiscale method.

5.3 Densely Sampled Field, Heterogeneous Model

A sample function of a nonstationary prior model is shown in Figure 10(a). The 64x64 pixels were divided into groups g_1 and g_2: g_1 contains the pixels in the upper left and lower right of the image, and g_2 contains the pixels in the diagonal band running through the center of the image. The prior model for g_1 is the "wood" model of Table 1; the prior model for g_2 uses the same coefficients in Table 1, but with the table rotated by 90 degrees. The cross correlation between groups g_1 and g_2 is zero. The choice of such a nonstationary prior, as opposed to the simple prior in the previous example, just implies a change in the prior statistics on the finest scale of the multiscale tree; the multiscale model development and estimation procedure proceed unaffected.

Figure 10(b) shows a noisy version of the original image corrupted by white Gaussian noise to 0dB; Figure 10(c) shows the corresponding multiscale reconstruction based on an overlapping model with $M = 8$ and $k = 32$. As we have stressed, the operator H_x removes blockiness without performing spatial blurring; thus the edge between the two regions g_1 and g_2 is well-preserved.

5.4 Locally Sampled Field, Homogeneous Model

We consider two final estimation problems involving a stationary prior model, but with measurements available at only non-rectangular subsets of the pixels. Figure 11(a) shows a subset of the pixels of the "wood" texture from Figure 8(a); this elliptical set of pixels represents those pixels to be used as measurements, implying a trivial change in the measurement projection operator G_y and in the multiscale measurement matrices on the finest scale of the tree. It is significant to note, however, that while the multiscale framework is readily adapted to irregular measurements, a change from dense to irregular sampling makes FFT-based approaches inapplicable.

Figure 11(b) shows the multiscale reconstruction based on the set of measurements given in Figure 11(a). The estimates capture the coarse features of the original texture of Figure 8(a) outside of the measured region, including certain aspects of the vertical bands to the left and right of the measured region. Also, once again, the estimated texture evolves smoothly, without blocky artifacts.

Figure 12 provides one additional illustration of irregular sampling: observations distributed according to a 2-D Poisson process. Figures 12(b),(c) display estimates based on a non-overlapping model of order 40 and

an overlapping model of order 16, both having the same computational load. The estimates computed by the overlapping model are more visually pleasing and also have a lower MSE.

6 Conclusions

We have presented a new approach to modeling and estimation using a recently introduced class of multiscale stochastic processes. Our work has been motivated by the observation that estimates based on the types of multiscale models previously proposed can exhibit a visually distracting blockiness. To eliminate this blockiness, we have discarded the standard assumption that distinct nodes on a given level of the multiscale process must correspond to disjoint portions of the image domain. Instead, we allow distinct tree nodes to correspond to overlapping portions of the image domain. This is done in a way that eliminates blocky artifacts *without* spatial averaging, so that if a field does have sharp discontinuities, these can be captured without blurring. By coupling this overlapping framework with a multiscale stochastic realization technique based on canonical correlations, we have developed a powerful estimation and modeling tool which allows one to manage the tradeoff among estimate smoothness, statistical fidelity, and computational effort.

The flexibility of the multiscale framework allows us to confront problems for which FFT techniques are not applicable; in particular, in problems involving nonstationary statistics or irregularly sampled data. In fact, the flexibility of our framework is greater than that implied by examples considered here; in particular, the modeling and estimation of processes in higher dimensions is also possible.

A Proof of Proposition 1

Using (19) and (4) we can derive

$$
\begin{aligned}
(CPC^T + R)[G_y^T(C_l P_l C_l^T + R_l)^{-1}G_y] &= [(RG_y^T R_l^{-1} G_y)CPC^T + R][G_y^T(C_l P_l C_l^T + R_l)^{-1}G_y] \\
&= [RG_y^T R_l^{-1} C_l P_l C_l^T + RG_y^T](C_l P_l C_l^T + R_l)^{-1}G_y \\
&= RG_y^T R_l^{-1}(C_l P_l C_l^T + R_l)(C_l P_l C_l^T + R_l)^{-1}G_y = I.
\end{aligned}
$$

We now verify (20). Using the above derivation with (4) and (10) leads to the following identities:

22

$$
\begin{aligned}
L &= PC^T(CPC^T + R)^{-1} = PC^T G_y^T \left(C_l P_l C_l^T + R_l \right)^{-1} G_y \\
&= PG_x^T C_l^T \left(C_l P_l C_l^T + R_l \right)^{-1} G_y \\
&= H_x P_l C_l^T \left(C_l P_l C_l^T + R_l \right)^{-1} G_y = H_x L_l G_y
\end{aligned}
$$

To verify (21), we use (11), (20), and (4) in the following sequence of identities:

$$
\begin{aligned}
\tilde{P} &= P - LCP = H_x P_l H_x^T - H_x L_l G_y CP \\
&= H_x P_l H_x^T - H_x L_l C_l G_x P \\
&= H_x \left(P_l - L_l C_l P_l \right) H_x^T = H_x \tilde{P}_l H_x^T.
\end{aligned}
$$

References

[1] H. Akaike. "Markovian Representation of Stochastic Processes by Canonical Variables." *SIAM Journal of Control* (13) #1, 1975.

[2] M. Basseville, A. Benveniste, K. Chou, S. Golden, R. Nikoukhah, and A. Willsky. "Modeling and Estimation of Multiresolution Stochastic Processes." *IEEE Transactions on Information Theory* (38), pp. 766-784, 1992.

[3] R. Chellappa and S. Chatterjee. "Classification of textures using Gaussian Markov random fields." *IEEE Transactions on ASSP*, (33), pp. 959-963, 1985.

[4] K. Chou, A. Willsky, A. Benveniste, "Multiscale Recursive Estimation, Data Fusion, and Regularization", *IEEE Trans. on Automatic Control* (39) #3, pp.464–478, 1994

[5] R. Coifman, D. Donoho, "Translation Invariant De-Noising," in *Lecture Notes in Statistics (#103): Wavelets and Statistics*, Springer-Verlag, New York, pp. 125–150, 1995

[6] P. Fieguth, A. Willsky, W. Karl, "Multiresolution Stochastic Imaging of Satellite Oceanographic Altimetric Data." *Proceedings of 1994 IEEE International Conference on Image Processing* (II), pp. 1-5, 1994.

[7] P. Fieguth, W. Karl, A. Willsky, C. Wunsch, "Multiresolution Optimal Interpolation and Statistical Analysis of TOPEX/POSEIDON Satellite Altimetry," *IEEE Trans. Geoscience and Remote Sensing* (33) #2, pp.280–292, 1995

[8] P. Fieguth, A. Willsky, W. Karl, "Efficient Multiresolution Counterparts to Variational Methods for Surface Reconstruction," submitted for publication

[9] P. Fieguth, *Application of Multiscale Estimation to Large Scale Multidimensional Imaging and Remote Sensing Problems*, PhD Thesis, Dept. of EECS, MIT, 1995

[10] Å. Björck and G. Golub. "Numerical Methods for Computing Angles Between Linear Subspaces." *Mathematics of Computation* (27) #123, 1973.

[11] H. Hotelling. "Relations between two sets of variates." In *Biometrika* (28), pp. 321-377. 1936.

[12] W. Irving, W. Karl, A. Willsky, "A Theory for Multiscale Stochastic Realization", *33rd Conference on Decision and Control*, 1994.

[13] W. Irving, A. Willsky, "A Theory for Multiscale Stochastic Realization of Gaussian Random Processes and Fields", submitted for publication

[14] B. Kosko (editor) *Neural Networks for Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, pp.37–61, 1992

[15] M. Luettgen, W. Karl, A. Willsky, "Efficient Multiscale Regularization with Applications to the Computation of Optical Flow." *IEEE Transactions on Image Processing* (3) #1, pp. 41-64, 1994.

[16] M. Luettgen, A. Willsky, "Multiscale Smoothing Error Models", *IEEE Trans. on Automatic Control* (40) #1, 1995

[17] M. Luettgen, W. Karl, A. Willsky, R. Tenney, "Multiscale Representations of Markov Random Fields", *IEEE Trans. Signal Processing* (41) #12, pp.3377–3396, 1993

[18] M. Luettgen, A. Willsky, "Likelihood Calculation for a Class of Multiscale Stochastic Models, with Application to Texture Discrimination.", *IEEE Trans. Image Processing* (4) #2, pp.194–207, 1995

[19] D.F. Morrison. *Multivariate Statistical Methods*. McGraw-Hill Book Company, NY, 1967.

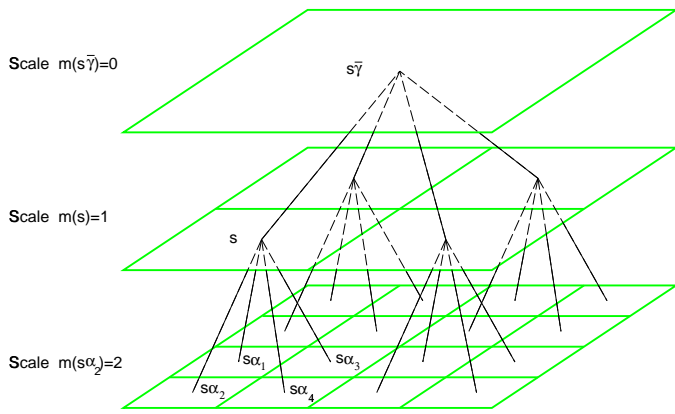[20] R.J. Muirhead. *Aspects of Multivariate Statistical Theory*. John Wiley & Sons, Inc., NY, 1982.

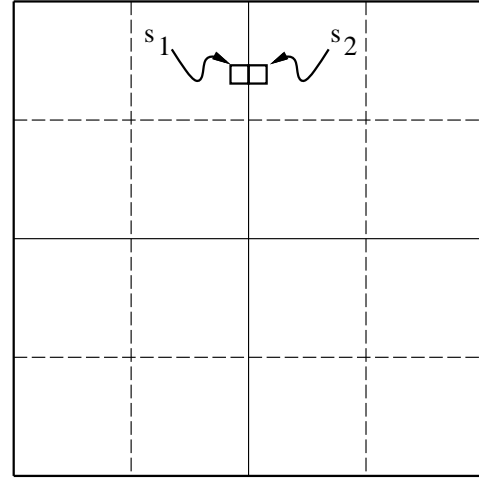Figure 1: Illustration of the first three levels of a quad-tree.

Figure 2: Two nodes, s_1 and s_2, neighbors in physical space, but distantly separated in tree space.
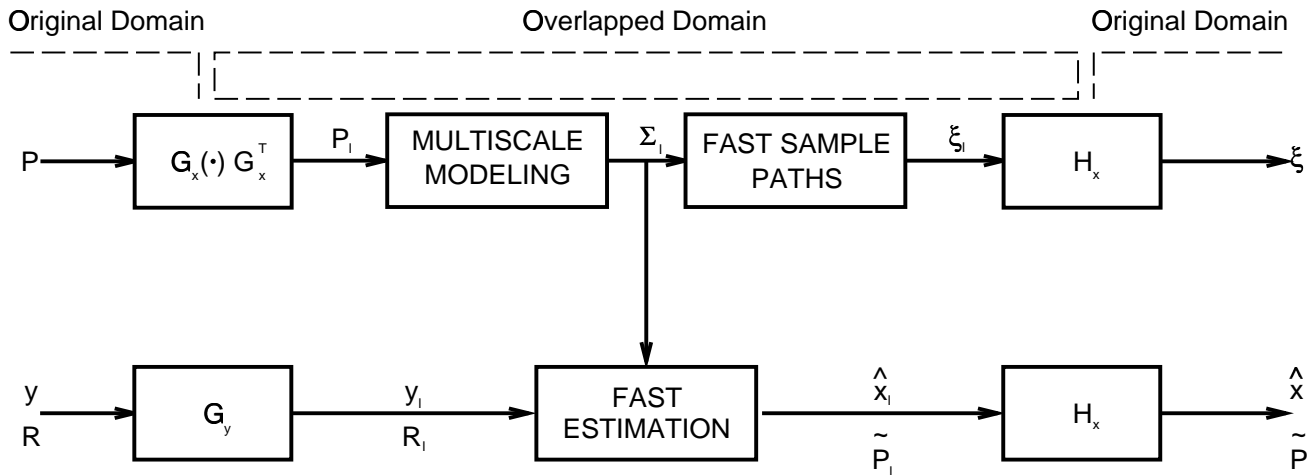
Figure 3: An abstract view of our overlapped approach to multiscale-based modeling and least-squares estimation. Fast multiscale estimation and sample-path generation are accomplished in the overlapped domain. G_x projects the statistics of x into the overlapped domain; G_y projects measurements y into the domain; and H_x, which possesses certain smoothness properties, projects the estimates $\hat{x}_l$ back out of the overlapped domain.
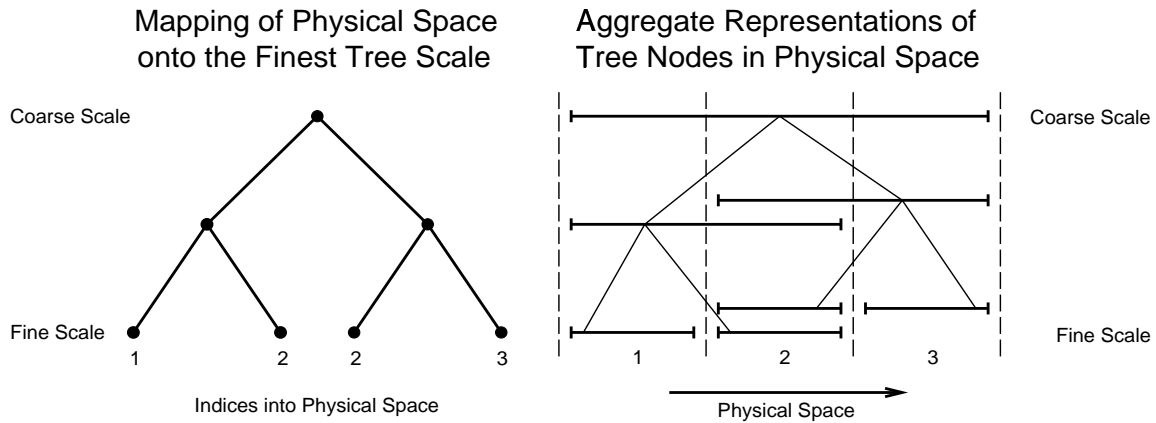
Figure 4: Illustration of an overlapping-tree representation of a process of length three, showing both the dyadic tree (left) on which the representation is based, and depiction (right) of the representation of each tree node. The bar ⊢⊣ associated with each tree node represents the subset of the points $\{1, 2, 3\}$ associated with that node.
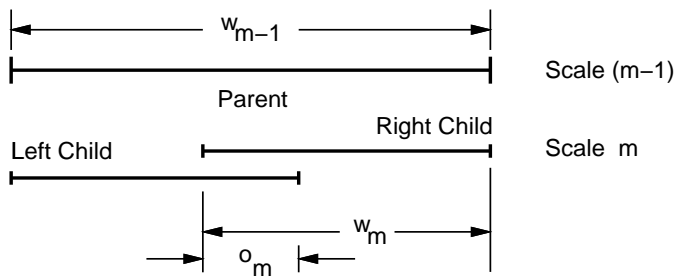

Figure 5: Basic overlapping-tree notation: o_m represents the degree of overlap between the regions represented by sibling multiscale nodes on scale m; w_m represents the width of the region represented by each node on scale m.
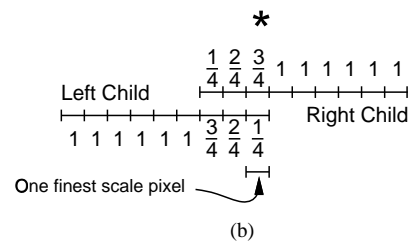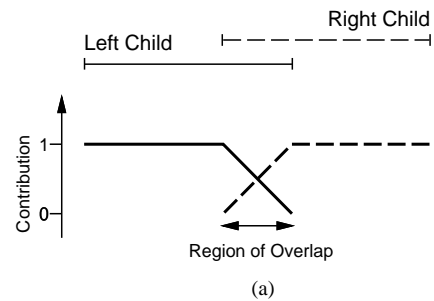

Figure 6: Two overlapping nodes: the set of relative contributions to each finest-scale pixel must sum to one. The contributions are tapered linearly over the region of overlap. Figure (a) shows this tapering pictorially; Figure (b) provides a specific example for two nodes which overlap by three pixels.
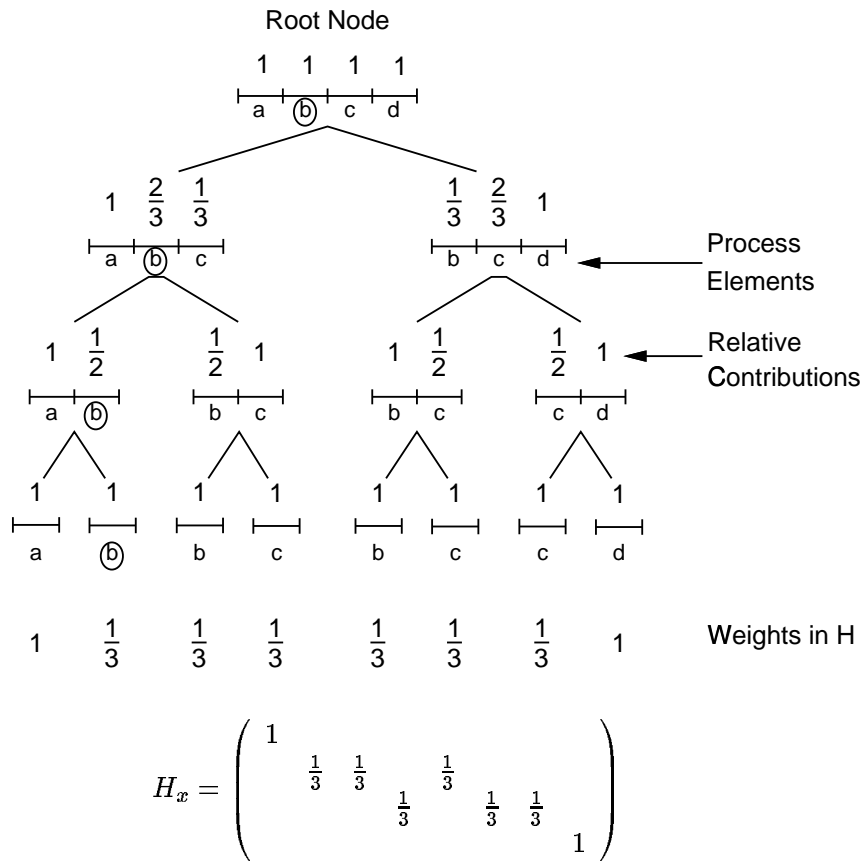
Figure 7: An example of the construction of H_x. A four-level tree is used to represent a process having four points (a, b, c, d). The process points associated with a multiscale node are indicated below the node. The relative contributions of each node to its associated process points are indicated above each node. Products of these relative contributions determine the elements of H_x.
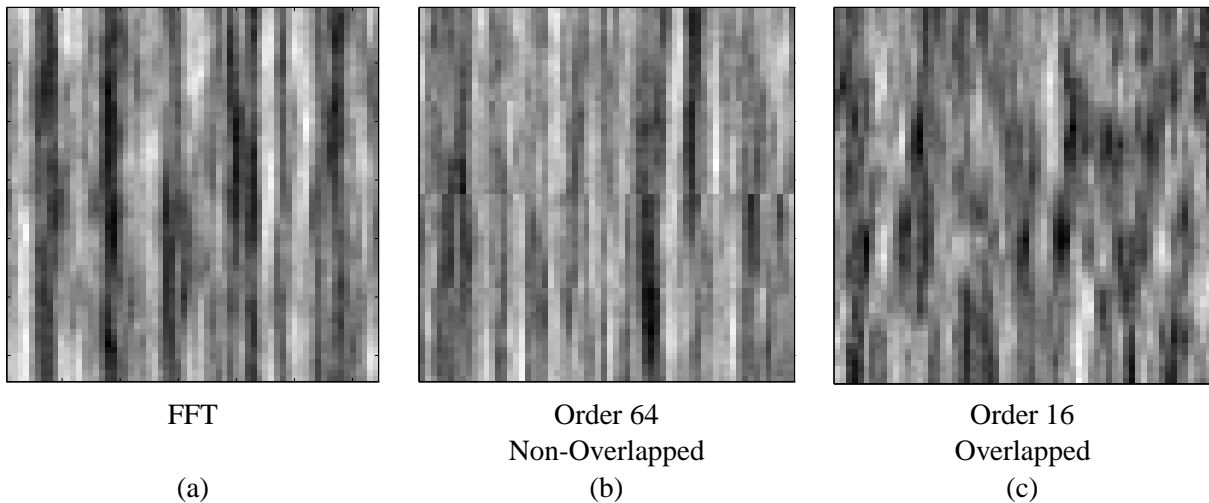

	FFT	Order 64 Non-Overlapped	Order 16 Overlapped
	(a)	(b)	(c)

Figure 8: Three simulated "wood" textures, 64×64 samples, based on an exact FFT approach in (a), and based on regular, multiscale trees in (b),(c).

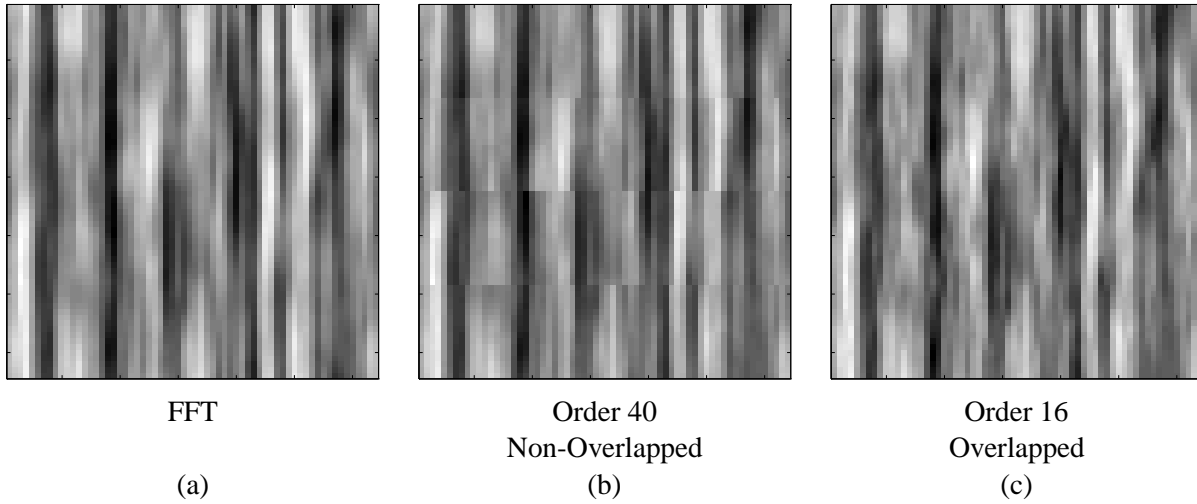| FFT	Order 40 Non-Overlapped	Order 16 Overlapped
(a)	(b)	(c)

Figure 9: Three estimated textures, each based on noisy measurements of Figure 8 (a). Estimate (a) is based on optimal FFT techniques, (b) is based on a non-overlapping tree of order 40, and (c) is based on an overlapping tree of order 16. The computational effort of the latter two estimates is the same, however note the artifacts visible across the quadrant boundaries in (b) which are completely removed in (c).

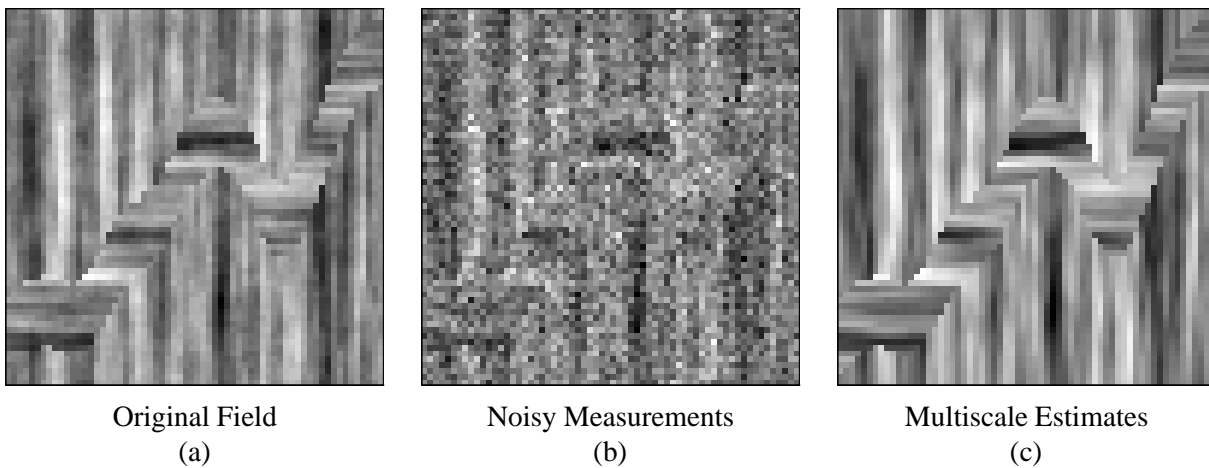| Original Field	Noisy Measurements	Multiscale Estimates
(a)	(b)	(c)

Figure 10: (a) shows a sample path of an inhomogeneous Markov random field, where each pixel belongs to a horizontally or vertically correlated texture. (b) shows the random field corrupted by 0dB white, Gaussian, noise. (c) shows the texture estimated using an inhomogeneous overlapped multiscale model, based on the measurements of (b) and given the correct prior texture model at each pixel.

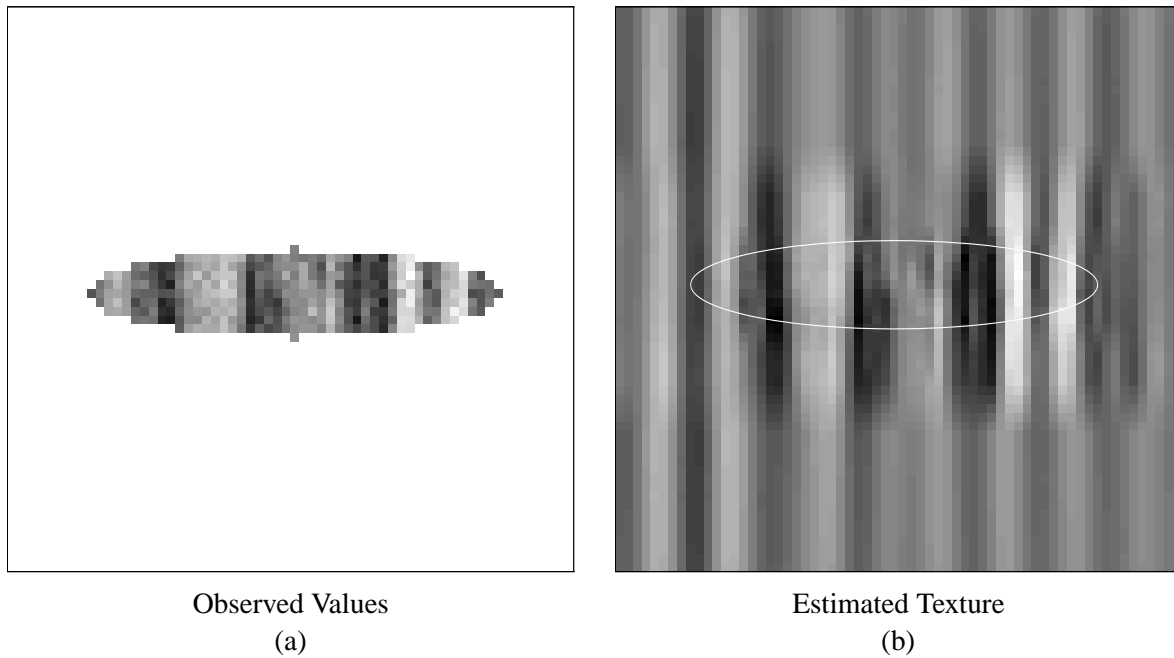Observed Values
(a)

Estimated Texture
(b)

Figure 11: The texture (b) was estimated using an overlapped multiscale model, based on the measurements (a) of a small subset of the wood texture. Despite the use of a multiscale estimator, the estimates evolve smoothly from the region in which measurements are present to the surrounding area without measurements.

Measurements
(a)

Estimates: Order 40
Non-overlapped
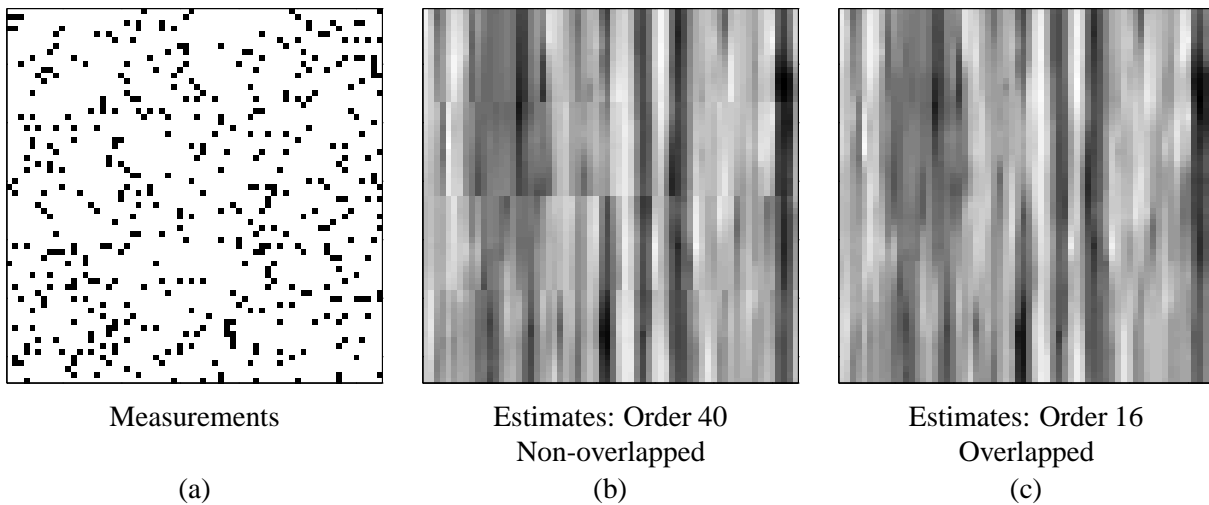(b)

Estimates: Order 16
Overlapped
(c)

Figure 12: Estimates produced by (b) a non-overlapping and (c) an overlapping tree, given measurements of a random subset of the pixels of Figure 8(a): one measurement is made at each black pixel in (a).