Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 197 (2008) 3492–3515

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

# Multiscale stochastic modeling for tractable inference and data assimilation ☆

Myung Jin Choi, Venkat Chandrasekaran, Dmitry M. Malioutov, Jason K. Johnson, Alan S. Willsky *

*Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

Received 20 August 2007; received in revised form 21 November 2007; accepted 21 December 2007
Available online 18 January 2008

**Abstract**

We consider a class of multiscale Gaussian models on pyramidally structured graphs. While such models have been considered in the past, very recent advances in inference methods for graphical models not only yield additional motivation for this class of models but also bring techniques that lead to new and powerful algorithms. We provide a brief summary of these recent advances – including so-called *walk-sum analysis*, methods based on Lagrangian relaxation, and a new method for "low-rank," wavelet-based, unbiased estimation of error variances – and then adapt and apply them to problems of estimation for pyramidal models. We demonstrate that our models not only capture long-range dependencies but that they also have the property that conditioned on neighboring scales, the correlation behavior within a scale is dramatically compressed. This leads to algorithms resembling multipole methods for solving partial differential equations in which we alternate computations across-scale (using an embedded tree in the pyramidal graph) with local updates within each scale. Not only are these algorithms guaranteed to converge to the correct answers but they also lead to new, adaptive methods for choosing embedded trees and subgraphs to achieve rapid convergence. This approach also leads to a solution to the so-called *re-estimation* problem in which we seek to update an estimate rapidly after local changes are made to the prior model or to the available data. In addition, by using a consistent probabilistic model across as well as within scales, we are able both to exploit low-rank variance estimation methods and to develop efficient iterative algorithms for parameter estimation.
© 2008 Elsevier B.V. All rights reserved.

*Keywords:* Graphical models; Multiresolution models; Statistical inference; Data assimilation

## 1. Introduction

In recent years the idea of using multiscale stochastic models for the purposes of statistical inference and estimation has received considerable attention, motivated not only by the efficiencies that multiscale representations may, in some cases, provide but also by the fact that either the phenomenon of interest, the available data, or the inference objectives involve behavior at multiple scales. Despite the already rich literature in this area (see, for example, [1] or many of the references cited here), there remains considerable motivation for further work. This paper describes some of our recent contributions to this important line of investigation.

As in much of the previous work, we focus on classes of so-called graphical models, i.e., Markov random fields defined on particular graphs, in which the nodes of the graph index a collection of random variables or vectors and the structure of the graph captures the dependency structure

☆ The research described in this paper was supported in part by Shell International Exploration and Production Inc., and in part by the Army Research Office under Grant W911NF-05-0-0207.

* Corresponding author. Tel.: +1 617 253 2356; fax: +1 617 258 8364.
*E-mail addresses:* myungjin@mit.edu (M.J. Choi), venkatc@mit.edu (V. Chandrasekaran), dmm@mit.edu (D.M. Malioutov), jasonj@mit.edu (J.K. Johnson), willsky@mit.edu (A.S. Willsky).

0045-7825/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2007.12.021

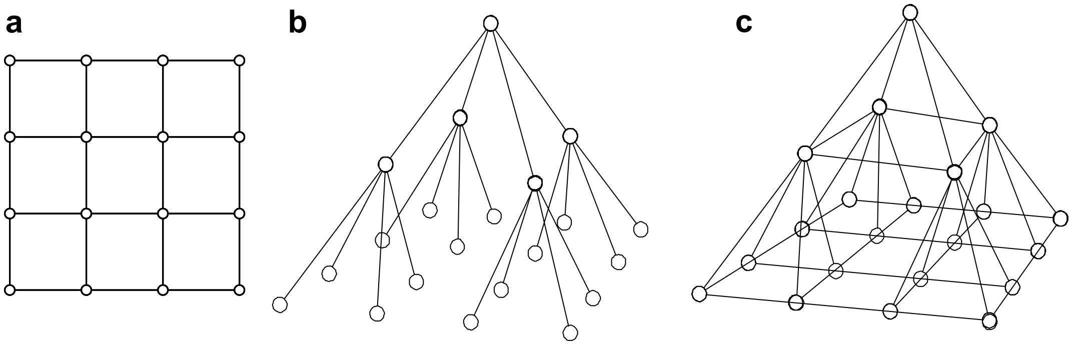


Fig. 1. Graphical models for a two-dimensional stochastic process: (a) lattice; (b) multiscale tree; and (c) pyramidal graph.

among these variables. Three prototypical examples of such graphs are those shown in Fig. 1. Estimation and inference problems of importance for models such as these include both data assimilation and model estimation. In the former we are given noisy measurements at some or all of the nodes and wish to compute both optimal estimates of the variables throughout the graph (or only at a selected subset of nodes) as well as the statistical structure (e.g., variances and correlations) of the errors in those estimates. In the latter we wish to use data to construct such a graphical model or to estimate its parameters. Both of these problems are easy for the pyramidal tree in Fig. 1b, as optimal inference has complexity that scales linearly with the number of nodes in the pyramid. However, optimal estimation is far more complex – often prohibitively so – for models on graphs as in Fig. 1a. The reason for this difference stems from the simple fact that the graph in Fig. 1b is a tree and hence has no loops (i.e., no sequence of distinct edges that form a cycle), while the graph in Fig. 1a has many loops.

This difference has sparked work in two directions, namely (i) building (typically approximate) models for problems of interest that involve multiscale trees as in Fig. 1b and (ii) developing inference methods for so-called loopy graphs, as in Fig. 1a that are both tractable (i.e., that yield algorithms with complexity that scales modestly with graph size) and as nearly optimal as possible. While there are certainly interesting questions that remain along the first of these lines of investigation (including our own recent work [2]), the same characteristic that makes trees so attractive computationally, namely the lack of loops, also limits their expressive power – e.g., one can find pairs of nodes at the finest scale of Fig. 1b that are spatial neighbors at that scale but that are connected on the graph only through a distant ancestor at a much coarser scale, a fact that can lead to blocky artifacts in both statistics and inference results. This limitation has been recognized by others [3–8] who have been motivated to consider graphs that retain the pyramidal structure of Fig. 1b but introduce connectivity among nodes at each level, as illustrated in Fig. 1c. The focus of our development in Section 3 is on Gaussian models with this type of pyramidal structure and on new classes of algorithms that exploit several recent advances in tractable inference.

In the next section, we introduce graphical models and statistical inference problems of interest for such models and summarize both some of the basic concepts associated with solving such problems and some recent advances on which we build in this paper. In particular, we briefly review why inference problems are easily solved on cycle-free graphs and are much more difficult in general on loopy graphs. As the Gaussian case is the focus of the new methods in this paper, we devote most attention to that case and review the recently introduced concept of *walk-summability* for Gaussian graphical models. We then briefly review several methods we have developed that exploit tractable substructure in complex graphs, methods that make explicit use of the concept of walk-summability.

Section 3 contains the new models, methods, and results developed in this paper. We begin by providing motivation for why models on pyramidal graphs represent a natural and principled choice for many applications, including those involving spatially extensive physical processes and fields. We then introduce the class of pyramidal models on which we focus and demonstrate that these models not only can capture long-scale dependencies but also have the important property that, when conditioned on neighboring scales, in-scale correlations are dramatically compressed. This leads to algorithms reminiscent of multipole methods for solving partial differential equations. Moreover, the results described in Section 2 both guarantee convergence and also lead to adaptive algorithms for rapid convergence. These adaptive methods also lead to very efficient methods for so-called *re-estimation* problems in which we wish to update an estimated field rapidly given local changes to model or data. Furthermore, we demonstrate how both the structure of our models and the advances summarized in Section 2 lead to other efficient algorithms for estimation, for the computation of bounds on and accurate approximations to error variances, and for parameter estimation for our new multiscale models. Finally in Section 4, we briefly discuss further research directions that use the ideas developed here as a point of departure.

## 2. Preliminaries

### 2.1. Graphical models

In this section, we introduce some basic notions for graphical models. For more details, we refer the reader to [9–13]. A *graph* $\mathscr{G} = (V, \mathscr{E})$ consists of a set of *vertices* or *nodes* $V$ and associated *edges* $\mathscr{E} \subset \binom{V}{2}$, where $\binom{V}{2}$ represents the set of all unordered pairs of vertices. An *edge* between nodes $i$ and $j$ is denoted by $\{i, j\}$. Two vertices are said to be *neighbors* if there is an edge between them. We use the notation $N(i)$ to denote the set of *neighbors* of node $i$, i.e., the nodes of the graph that are each connected to $i$ by an edge. The cardinality of $N(i)$ is referred to as the *degree* of node $i$. A *subgraph*[1] $\mathscr{S}$ of $\mathscr{G} = (V, \mathscr{E})$ is any graph whose vertex set is $V' \subseteq V$, and whose edge set $\mathscr{E}'$ is a subset $\mathscr{E}' \subseteq \mathscr{E}(V')$, where

$$\mathscr{E}(V') \triangleq \{\{i, j\} \mid \{i, j\} \in \mathscr{E}, i, j \in V'\}. \tag{1}$$

A subgraph is said to be *spanning* if $V' = V$. An *induced subgraph* $\mathscr{S}(V')$ is a subgraph with vertices $V'$ and edges $\mathscr{E}' = \mathscr{E}(V')$. A *supergraph* $\mathscr{H}$ of $\mathscr{G}$ is any graph whose vertex set $V'$ is a superset $V' \supseteq V$, and whose edge set $\mathscr{E}'$ is a superset $\mathscr{E}' \supseteq \mathscr{E}$. A *path* $u_0 \cdots u_\ell$ between two vertices $u_0$ and $u_\ell$ in $\mathscr{G}$ is a sequence of distinct vertices $\{u_k\}_{k=0}^{\ell}$ such that there exists an edge between each successive pair of vertices, i.e., $\{u_k, u_{k+1}\} \in \mathscr{E}$ for $k = 0, \ldots, \ell - 1$. A subset $S \subset V$ is said to *separate* subsets $A, B \subset V$ if every path in $\mathscr{G}$ between any vertex in $A$ and any vertex in $B$ passes through a vertex in $S$. A graph is said to be *connected* if there exists a path between every pair of vertices. A *clique* is a fully connected subgraph, i.e., a subgraph in which each vertex is linked to every other vertex by an edge. A clique is *maximal* if it is not contained as a proper subgraph of any other clique. A *cycle* is the concatenation of a path $u_0 \cdots u_k$ with the vertex $u_0$ such that $\{u_k, u_0\} \in \mathscr{E}$. A *tree* is a connected graph that contains no cycles. A graph is said to be *chordal* or *triangulated* if every cycle of length greater than three in the graph contains an edge between non-neighboring vertices in the cycle. A special representation for a chordal graph can be specified in terms of the maximal cliques of the graph. Let $\mathscr{C}$ be the set of maximal cliques in a connected graph $\mathscr{G}$. A *junction tree* representation of $\mathscr{G}$ is a tree, with the nodes being the elements of $\mathscr{C}$, which satisfies the following *running intersection* property: for every pair of nodes (cliques) $C_i$ and $C_j$ in the junction tree, every node (clique) in the unique path between $C_i$ and $C_j$ contains $C_i \cap C_j$. Valid junction trees can only be defined for chordal graphs [11]. A graph $\mathscr{G}$ is said to be *thin* if the smallest chordal supergraph of $\mathscr{G}$ (i.e., one with the least number of extra edges) has small maximal cliques.

A *graphical model* [11–13] is a collection of random variables indexed by the vertices of a graph[2] $\mathscr{G} = (V, \mathscr{E})$; each vertex $i \in V$ corresponds to a random variable $x_i$, and where for any $A \subset V$, $x_A \equiv \{x_i \mid i \in A\}$. A distribution $p(x_V)$ is *Markov* with respect to a graph $\mathscr{G} = (V, \mathscr{E})$ if for any subsets $A, B \subset V$ that are separated by some $S \subset V$, the subset of variables $x_A$ is conditionally independent of $x_B$ given $x_S$, i.e., $p(x_A, x_B \mid x_S) = p(x_A \mid x_S) \cdot p(x_B \mid x_S)$. In this manner, graphical models generalize the concept of Markov chains, and are thus also referred to as *Markov random fields* (MRFs). A distribution being Markov with respect to a graph implies that it can be decomposed into local functions in a very particular way. Specifically, the Hammersley–Clifford Theorem [12] states that a sufficient condition for Markovianity that is also necessary for strictly positive probability distributions is that the joint distribution for $x \equiv x_V$ be expressible as a product of terms each of which is a function that depends only on the variables in a clique of the graph. For strictly positive distributions each of these terms can be represented as an exponential of a so-called *potential function*, so that the overall distribution belongs to an *exponential family* of distributions. Such a distribution can be written in the form

$$p_\theta(x) = \exp\{\theta^{\mathrm{T}}\phi(x) - \Phi(\theta)\}, \tag{2}$$

where $\phi(x)$ is a vector of *features* or *statistics*, each of which depends only on the variables in a single clique of the graph; $\theta$ is a vector of parameter coefficients; and $\Phi(\theta)$, which is known as the *log-partition function*, provides the normalization so that the distribution has unit total mass.[3] The log-partition function has many very important properties and deep connections to problems of inference for graphical models. Of importance to us here are its critical role in parameter estimation and the fact that it is a convex function.

For this paper, we focus on Gaussian distributions which are examples of so-called *pairwise* models in which the elements of $\phi(x)$ are all functions of variables at individual nodes or pairs of variables connected by edges. Such a model, which is also known as a Gauss–Markov Random Field (GMRF), is commonly thought of as being parameterized by a mean vector $\mu$ and a symmetric, positive-definite covariance matrix $P$ which we denote by $x \sim \mathscr{N}(\mu, P)$[4]:

$$p(x) = \frac{1}{(2\pi \cdot \det P)^{\frac{|V|}{2}}} \exp\left\{-\frac{1}{2}(x-\mu)^{\mathrm{T}}P^{-1}(x-\mu)\right\}. \tag{3}$$

[1] While it is a bit redundant, a subgraph is sometimes referred to as an *embedded subgraph* to emphasize that every node and edge in this subgraph can be found in the original graph in which it is embedded.

[2] The models that we consider are defined with respect to undirected graphs; we note that models defined on directed graphs can be converted to models on undirected graphs with some loss in structure [14].

[3] An implicit assumption here is that the parameter vector $\theta$ be constrained to the set for which $\Phi(\theta) < \infty$. For Gaussian models this reduces simply to the constraint that the covariance matrix be positive-definite.

[4] For simplicity in this paper we assume that each $x_s$ is a scalar random variable. It is a straightforward extension to allow each node to have a vector random variable.

An alternate natural parameterization for GMRFs is specified in terms of the *information matrix* $J = P^{-1}$ and *potential vector* $h = P^{-1}\mu$, and is denoted by $x \sim \mathcal{N}^{-1}(h, J)$:

$$p(x) \propto \exp\left\{-\frac{1}{2}x^{\mathrm{T}}Jx + h^{\mathrm{T}}x\right\}. \tag{4}$$

This is known as the *information form* representation. Using (4) and the Hammersley–Clifford Theorem, we see that there is a direct tie between the sparsity of $J$ and the Markov structure of $x$. Specifically, $x$ is Markov with respect to $\mathcal{G} = (V, \mathcal{E})$ if and only if $J_{ij} = 0$ for every $\{i,j\} \notin \mathcal{E}$. Indeed the elements of $J$ are also related to so-called *partial correlation coefficients*. Specifically the correlation coefficient between $x_i$ and $x_j$ conditioned on knowledge of all the other variables is given by [12]

$$\rho_{i,j} \triangleq \frac{\mathrm{cov}(x_i; x_j \mid x_{\setminus i,j})}{\sqrt{\mathrm{var}(x_i \mid x_{\setminus i,j})\mathrm{var}(x_j \mid x_{\setminus i,j})}} = -\frac{J_{ij}}{\sqrt{J_{ii}J_{jj}}}. \tag{5}$$

Hence, $J_{ij} = 0$ implies $x_i$ and $x_j$ are conditionally independent given all other variables $x_{\setminus i,j}$.[5]

Comparing the information form in (4) to the form of an exponential family in (2) we see that one natural way in which to define the vector of features $\phi$ is as

$$\phi(x) = \left(\begin{pmatrix} x_i^2 \\ x_i \end{pmatrix}, \forall i \in V\right) \cup \left(x_i x_j, \forall \{i,j\} \in \begin{pmatrix} V \\ 2 \end{pmatrix}\right), \tag{6}$$

which yields $\theta$ and $\eta$ parameters that are, respectively, given by elements of the $(J, h)$ and $(P, \mu)$ representations:

$$\theta = \left(\begin{pmatrix} -\frac{1}{2}J_{ii} \\ h_i, \end{pmatrix}, \forall i\right) \cup (-J_{ij}, \forall \{i,j\}), \tag{7}$$

$$\eta = \left(\begin{pmatrix} P_{ii} \\ \mu_i \end{pmatrix}, \forall i\right) \cup (P_{ij}, \forall \{i,j\}). \tag{8}$$

These "dual" parameterizations point out the crux of the estimation problem. The natural way in which many inference problems are specified – e.g., ranging from time series models and Kalman filtering problems to so-called thin-plate and thin membrane models described in Section 3 – is in terms of the information parameters; the challenge of estimation, then, is the computation of the moment parameters, a process that can, at least conceptually, be solved via matrix inversion (to recover $P$ from $J$) and solving the linear equations $J\mu = h$ to compute $\mu$ from $h$.

Graphical models arise naturally when we employ the principle of *maximum entropy* to construct models based on data or on partial specifications of the overall probabilistic structure of variables of interest. In particular, suppose that we are given the moments $\eta = E\{\phi(x)\}$ corresponding to a set of features. Then, assuming that these moments are consistent – i.e., that there is *some* distribution that can match them – then the maximum entropy solution is precisely of the form given in (2), i.e., the maximum entropy model is Markov with respect to the graph corresponding to these features and moments. A generalization of this result applies to the case in which the specified moments are inconsistent – i.e., when there is no distribution exactly matching all of the moments. In this case, it makes sense to relax the constraint of exactly matching the moments by allowing some tolerance for each. The *Maximum Entropy Relaxation* approach developed in [15] solves such a problem, resulting in a model in the same exponential family as in (2), but possibly with one or more of the edges (i.e., components of $\theta$ and $\phi(x)$) removed.

### 2.2. Statistical inference in graphical models

Consider the problem of estimating $x$ given data, $y$, each component of which is a possibly noisy measurement of the value of $x$ at a single node or on a clique of the graph, where the noises on these individual measurements are mutually independent. Thanks to Bayes' rule, the conditional distribution for $x$ given $y$ also has the same graphical structure, with modified coefficients depending on the observed measurement values.[6] Thus the problem of estimating $x$ based on $y$ involves computations on a model of the form of (2) where the dependence on the observed measurement values has been absorbed into the parameterization. In the Gaussian case, this is quite simple to describe. Specifically, if we have observations of the form:

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

where $v$ is zero-mean, Gaussian, with covariance $S$, then the information parameterization for the conditional distribution for $x$ given $y$ involves replacing the prior parameters $h$ and $J$ with

$$h + C^{\mathrm{T}}S^{-1}y, \tag{10}$$

$$J + C^{\mathrm{T}}S^{-1}C, \tag{11}$$

respectively. Note that if $C$ is a selection matrix – i.e., one with only a single non-zero entry per row (so that each measurement is of a single component of $x$) – and if $S$ is diagonal, then the sparsity structure of the information matrix is unchanged by conditioning. For the development and illustration of our new multiscale algorithms we will, indeed, assume that $S$ is diagonal.

There are at least two natural notions of estimation that are widely considered, namely the computation of the marginal statistics, i.e., the moments, $\eta$ (which requires computing the marginal distribution at individual nodes or edge-pairs), and the computation of the MAP estimate, i.e., the value of $x$ corresponding to the peak of the joint distribution. For GMRFs the MAP estimate is also the mean, so that we focus our discussion here on the first problem, namely that of

[5] We use the notation $x_{\setminus S}$ to denote the collection of all components of $x$ other than those indexed by nodes in $S$.

[6] In some problems, one or more of the measurements may be of a function of a set of variables not forming a clique in the original graph. This is accommodated simply by augmenting the graph with edges in order to make this set a clique. That is, the graph used for statistical inference captures all dependencies in either the phenomenon itself or in the observed data.

computing the marginal statistics at individual nodes in the graph. This estimation problem can be solved very efficiently – with complexity linear in $|V|$ – for graphs without loops or cycles, i.e., for trees such as in Fig. 1b, using a variety of algorithms that collectively go under the name of *Belief Propagation* (BP) [14]. BP is a generalization of the recursive forward–backward algorithms originally developed for the special case of inference in Markov chains [16]. BP has an interpretation as a "message-passing" algorithm in that messages are passed locally between variables along the edges of the tree, thus providing an efficient method to compute exact estimates in a distributed manner.

For graphs without loops, single nodes act as separators – i.e., conditioned on the value at any particular node, say $i$, the sets of variables in the several disconnected subtrees that result if node $i$ is removed are mutually independent. As a result, the marginal probability at a single node, say $i$, is proportional to a product of the local "evidence" at that node (corresponding to the potential function at that single node) and likelihoods from each of its neighbors. The likelihood provided by node $j$ to node $i$ can be thought of as a "message", capturing all of the evidence, relevant to the marginal at node $i$, in the subtree rooted at node $j$ and extending away from $i$. Furthermore, this likelihood can itself be decomposed in terms of likelihoods (messages) from all of the neighbors of node $j$ *other than* $i$, the local evidence at node $j$, and then a "transition" of this information from node $j$ to $i$ using the edge potential between these two nodes. These likelihood equations then form a set of coupled, fixed-point equations which can be iterated to convergence. The choice of which messages are updated at each stage of the iteration is quite flexible, ranging from a completely parallel scheme to serial implementations. For example, some arbitrary node $i \in V$ can be assigned as the "root" node, and messages can be passed in an up–down sweep from leaves (degree-1 nodes) to the root and back to the leaves. This is precisely the form often used to describe optimal estimation algorithms for multiresolution trees such as in Fig. 1b. Whatever scheme is used for updating or *scheduling* messages in a tree-structured graph, BP iterations converge to the correct likelihoods, in a finite number of iterations proportional to the diameter of the graph (length of the longest path) [14].

In tree-structured Gaussian graphical models with $x|y \sim \mathcal{N}^{-1}(h, J)$, we obtain the following parametric BP updates [17,18]:

$$J_{j\backslash i}^{(n)} = J_{jj} + \sum_{u \in N(j)\backslash i} \Delta J_{u\to j}^{(n-1)}; \quad h_{j\backslash i}^{(n)} = h_j + \sum_{u \in N(j)\backslash i} \Delta h_{u\to j}^{(n-1)}, \tag{12}$$

$$\Delta J_{j\to i}^{(n)} = -J_{ij} J_{j\backslash i}^{(n)^{-1}} J_{ji}; \quad \Delta h_{j\to i}^{(n)} = -J_{ij} J_{j\backslash i}^{(n)^{-1}} h_{j\backslash i}^{(n)}, \tag{13}$$

$$J_i^{(n)} = J_{ii} + \sum_{j \in N(i)} \Delta J_{j\to i}^{(n)}; \quad h_i^{(n)} = h_i + \sum_{j \in N(i)} \Delta h_{j\to i}^{(n)}. \tag{14}$$

At iteration $n$, the estimates for the mean and variance at node $i$ are computed as $\mu_i^{(n)} = J_i^{(n)^{-1}} h_i^{(n)}$ and $P_{ii}^{(n)} = J_i^{(n)^{-1}}$. On convergence, these estimates yield the exact values of the marginal means and variances, yielding an algorithm with total complexity proportional $|V|$. This tremendous savings in computing the diagonal elements of the inverse of $J$ and solving the estimation equations $J\mu = h$ stem from the very special sparsity structure of $J$ for a tree-structured graph: in such a case, there are *elimination orders*, i.e., orders for Gaussian elimination, which induce *no fill* as variable elimination proceeds.

For graphs with loops, exact estimation algorithms are considerably more complex. In particular, because single nodes do not in general form separators for graphs with loops, we lose the conditional independence properties that allow us to factor probabilities and likelihoods. In particular, for GMRFs, in using Gaussian elimination to solve for the variances or solving the equations for the means, we induce fill after variable elimination. For example, eliminating a column of nodes in the graph of Fig. 1a results in a graphical model for the remaining nodes in which there are dense connections between the nodes in the columns immediately to the left and right of the column eliminated. Such an exact approach to inference corresponds to grouping nodes together in order to form a so-called *junction tree* (to which exact tree-based inference can be applied). However, as for graphs such as Fig. 1a, the number of nodes that need to be grouped together can be quite large (thanks to fill), rendering such direct methods ineffective.

Developing estimation algorithms on loopy graphs represents an active area of continuing research. One approximate method that has been widely used is *Loopy Belief Propagation* (LBP) [19] which simply involves iterating the fixed-point equations despite the fact that they are no longer valid when the graph in question has loops. LBP may or may not converge and if it does converge the results it produces may or may not be good approximations to the desired probabilities and moments.

For random fields with graphical structure as in Fig. 1a several different algorithms have been developed that correspond to approximating the statistics of this field by multiresolution models on a tree as in Fig. 1b and then performing exact inference on this approximate model. These approaches are based on construction of a particular junction tree for the model in Fig. 1a obtained by a divide and conquer approach – e.g., chopping the field up into smaller and smaller subregions and using the set values of the field around the boundary of each of these subregions as the state at one of the nodes in the multiresolution tree. As these boundaries grow in size for larger regions, exact inference for the resulting tree model is itself intractable, requiring approximations. We refer the reader to [1,20] for an early approach to making such approximations and to [2,21] for a very recent method with considerably enhanced performance.

In this paper, we make use of several advances in inference for Gaussian graphical models to develop high-performance algorithms for the richer class of Gaussian models defined on graphs as in Fig. 1c. The next three subsections provide brief summaries of these recent advances.

### 2.3. Walk-sum interpretation of Gaussian estimation

An insightful concept for both understanding and analyzing inference algorithms for GMRFs is that of *walk-sum analysis* [18,22]. For simplicity only we assume in this section that the matrix $J$ has been normalized to have unit diagonal entries,[7] so that $J = I - R$, where $R$ is precisely the matrix of partial correlation coefficients in (5). As a result, the elements in any power of this matrix correspond to sums of weighted *walks* along paths in the graph $\mathcal{G}$ where the partial correlation coefficients, $R_{ij}$, and their products provide the weights.

Specifically, a *walk* in $\mathcal{G}$ is a sequence of vertices $w = \{w_k\}_{k=0}^{\ell}$ such that $\{w_k, w_{k+1}\} \in \mathcal{E}$ for each $k = 0, \ldots, \ell - 1$. The *weight* of the walk $\phi(w)$ is defined

$$\phi(w) \triangleq \prod_{k=0}^{\ell-1} R_{w_k, w_{k+1}}.$$

It is then readily seen that $(R^{\ell})_{ij}$ equals the walk-sum $\phi(i \xrightarrow{\ell} j)$, i.e., the sum of the weights of all of the walks in the (finite) set of all length-$\ell$ walks from $i$ to $j$ [18]. Consequently, we have that the elements of the covariance matrix $P$ can be expressed and computed as

$$P_{ij} = ((I - R)^{-1})_{ij} = \sum_{\ell=0}^{\infty} (R^{\ell})_{ij} = \sum_{\ell=0}^{\infty} \phi(i \xrightarrow{\ell} j) \triangleq \phi(i \to j), \tag{15}$$

where $\phi(i \to j)$ denotes the walk-sum of *all* walks from $i$ to $j$. Moreover, the mean can be computed in terms of these walk-sums using the elements of $h$ as weights:

$$\mu_j = \sum_{i \in V} P_{ji} h_i = \sum_{i \in V} h_i \phi(i \to j). \tag{16}$$

Of course, these expressions are valid only if the infinite sums of walks are well defined. Since different algorithms – e.g., corresponding to different message-passing schedules – may compute these terms in different orders, we are led directly to the concept of *walk-summability*, namely the requirement that all of these sums converge absolutely. Walk-summability is an easily checked condition, namely the spectral radius of the matrix $\overline{R}$, in which every element of $R$ is replaced by its absolute value, is less than 1. While there are valid GMRFs that are not walk-summable, there are also large and important classes that are (including all GMRFs on trees as well as those introduced in Section 3). Walk-sum analysis has led to several important results (e.g., in the analysis of LBP) [18,23] one aspect of which we introduce in the next subsection and exploit in the multiresolution algorithms developed in this paper.

### 2.4. Algorithms that exploit tractable subgraphs

A very important concept employed in a number of advanced algorithms is that of exploiting exact inference algorithms on tractable subgraphs of a graphical model. In this section we briefly review two such approaches that we exploit in later sections. The first of these involves serial iterations using *embedded subgraphs* [23,24]. The general idea behind this approach is that in each iteration we choose a subset of the variables to be updated and a subset of the edges of the graph to be enforced. For example, at one extreme we might choose to update all of the nodes of a graphical model but enforce only those edges corresponding to an embedded *spanning tree* (or, more generally, a tractable subgraph – e.g., one with tractable junction tree) of the full graph. This corresponds to rewriting the equation for the moments as

$$J\mu = (J_{\text{tree}} - K)\mu = h, \tag{17}$$

where $K$ includes terms “cut” to expose the tree captured by $J_{\text{tree}}$. Rewriting this as

$$J_{\text{tree}}\mu = h + K\mu, \tag{18}$$

we have the starting point for the so-called *Embedded Trees* (ET) algorithm in which at each iteration we substitute in the current estimate of the mean $\mu$ on the right-hand side of (18) and solve the resulting equation to yield our next estimate of $\mu$. Alternatively, we can consider a Gauss–Seidel iteration in which we choose a subvector, $\mu_1$, of the components of $\mu$ to update, leaving the remaining components, $\mu_2$, fixed, where the only requirement is that solving for $\mu_1$, with all other components fixed in (17) is tractable. Finally, one can imagine hybrids of these in which we both choose a subset of variables to update (leaving all others fixed) and then use embedded tree-like iterations to compute the desired updates.

There are two very natural questions associated with this very rich class of algorithms, namely the issue of convergence and the method by which subgraphs are chosen at each successive iteration. Both of these are addressed in [23] with the following important results. First, for walk-summable models, the *only* requirement for convergence of any such embedded subgraph algorithm is that every node and every edge be included infinitely often in the iteration sequence – in this way guaranteeing that every walk in the walk-sum (16) is eventually included. As for the issue of the choice of each successive subgraph, [26] provides an adaptive procedure aimed at choosing an embedded subgraph in order to reduce the principle errors in the preceding iterations. Specifically, let $\hat{x}^{(n)}$ denote the estimate after the $n$th iteration, and let $h^{(n)} = h - J\hat{x}^{(n)}$ denote the residual error. For each edge $\{i, j\}$ in $\mathcal{G}$ we then compute a weight:

$$\omega_{ij} = \frac{\mid \rho_{i,j} \mid}{1 - \mid \rho_{i,j} \mid} \cdot (\mid h_i^{(n-1)} \mid + \mid h_j^{(n-1)} \mid). \tag{19}$$

This weight captures the reduction in residual error (from iteration $n - 1$ to $n$) if we were to take our next embedded

[7] If $D$ is a diagonal matrix containing the diagonal entries of $J$, then the matrix $D^{-\frac{1}{2}}JD^{-\frac{1}{2}}$ contains re-scaled entries of $J$ at off-diagonal locations and 1's along the diagonal.

subgraph simply to be the nodes $i$ and $j$ and the edge connecting them. As shown in [23] minimizing an upper bound on the error $x - \hat{x}^{(n)}$ is equivalent to solving the max-weight spanning tree problem

$$\arg \max_{S_n \text{ a tree}} \sum_{\{i,j\} \in S_n} \omega_{ij}. \tag{20}$$

Solving this tractable problem yields a spanning tree to be used in the next iteration.

As an alternative to updating all nodes using a spanning tree, we can construct a version of Gauss–Seidel in which we adaptively choose a subset of variables to update at each iteration, leaving the remaining variables unchanged. An efficient algorithm to do this is described in [23], and we refer the reader to that paper for details. The basic idea is as follows: we initially assign weights to every node in the graph equal to the absolute value of that nodes residual error, i.e.:

$$\omega_i = \mid h_i^{(n-1)} \mid . \tag{21}$$

At each stage in the selection process we examine the weights on all nodes that have not yet been selected for update at the next iteration and add to that set the node, call it $i^*$, with largest weight. We then adjust the weights of each of the remaining unselected nodes to account for the fact that walk-sums between any pair of selected nodes will also be accounted for in the next Gauss–Seidel iteration. Specifically, right after node $i^*$ has been selected, we adjust the weight of each neighbor, $j$, of $i^*$ that has not been selected as follows:

$$\omega_j \leftarrow \omega_j + \left( \mid h_{i^*}^{(n-1)} \mid + \mid h_j^{(n-1)} \mid \right) \frac{\mid \rho_{i^*,j} \mid}{1 - \mid \rho_{i^*,j} \mid}. \tag{22}$$

We then alternate steps of selecting a single node and updating the weights of unselected nodes until the set of selected nodes reaches a prespecified size.

The second approach to exploiting embedded tractable subgraphs is that of solving a set of tractable inference problems on such graphs *in parallel*, but subject to the constraint that the estimates produced by all of these subgraphs *agree* (i.e., produce a single, consistent estimate). The algorithm starts with a decomposition of $\mathscr{G}$ into tractable subgraphs $\mathscr{G}^k = (V^k, \mathscr{E}^k)$ and an initial valid decomposition of $J$ and $h$:

$$h = \sum_k h^k, J = \sum_k J^k \quad \text{and} \quad J^k \succ 0 \text{ for all } k. \tag{23}$$

While the general approach in [25] allows the subgraphs to share edges as well as nodes, we limit our discussion here to the case in which the subgraphs only share nodes.

Note that the mean of a Gaussian as in (3) is the same as finding the peak of the distribution or equivalently, maximizing the exponent in (4). Consider solving a set of separate maximizations, one for each $(J^k, h^k)$, but with the *equality constraint* that the resulting "estimates" all be equal. Using Lagrange multipliers to adjoin these equality constraints leads to a set of necessary and sufficient conditions that can be solved via an intuitively appealing iterative, *Lagrangian Relaxation* algorithm in which we alternately solve inference problems on the individual graphs and then modify the decomposition in (23) to force equality of estimates and variances at a single node. Of course after such an exchange, when we perform inference again on each of the separate graphs we will in general not have equality at any *other* node. Thus, we iteratively cycle through all of the nodes of the original graph – in any order, just as long as each node is revisited during each cycle. As shown in [25], this algorithm is guaranteed to converge to the optimal estimates for our original problem. The error covariances computed by this algorithm will not be correct; however, thanks to the convexity of the log-partition function, they do provide upper bounds. Moreover, again exploiting convexity, we can make these upper bounds as tight as possible. This is closely related to the so-called tree-reweighted belief propagation [26–28], but the Lagrangian Relaxation algorithm is specified for Gaussian models and also provides optimal choice of weights $\lambda_k$ for each subgraph as described below. We refer the reader to [25,29] for details and simply state the resulting algorithm here.

- Given the current decomposition (23), we perform inference on each graph, yielding, for the $k$th graph, estimates, $\hat{x}_j^k$, and error covariances, $P_j^k$, for every node, $j$ in[8] $\mathscr{G}^k$, and compute the corresponding information parameters $\widehat{J}_j^k = (P_j^k)^{-1}$ and $\hat{h}_j^k = \widehat{J}_j^k \hat{x}_j^k$.
- Choose a node $i$ in the original graph and for every graph $\mathscr{G}^k$ in which node $i$ appears modify the potential at that node as follows:[9]

$$J_{ii}^k \leftarrow J_{ii}^k + \left( \lambda_k \left( \sum_k \widehat{J}_i^k \right) - \widehat{J}_i^k \right), \tag{24}$$

$$h_i^k \leftarrow h_i^k + \left( \lambda_k \left( \sum_k \hat{h}_i^k \right) - \hat{h}_i^k \right), \tag{25}$$

where the non-negative weights $\lambda_k$ sum to one and are chosen to make the bounds on the log-partition function (and variance estimates) as tight as possible. Specifically, using the current decomposition (23), we compute these weights as follows:

$$\lambda_k = \frac{1}{Z} \exp \left\{ \frac{1}{N} \log \det J^k \right\}, \tag{26}$$

$$Z = \sum_k \exp \left\{ \frac{1}{N} \log \det J^k \right\}. \tag{27}$$

The computation of log det $J^k$ can be done efficiently as along as $J^k$ corresponds to a tractable subgraph. Note that it is easily checked that the first two conditions in (23) are

[8] It is not necessary that every node appear in every graph, e.g., as in Section 4 for our pyramidal graph.

[9] There is a simplified version of the algorithm that only updates $h^k$ to achieve agreement of means (for a fixed set of $J^k$), which would be sufficient to obtain the optimal estimates. However, we also optimize the splitting of $J$ into $J^k$ because it should help make the linear system for the means better conditioned (for faster convergence) and has the added benefit of leading to upper-bounds on the variances.

satisfied. The positive-definiteness of the modified $J^k$ is also maintained (see [29]).

- Note that when inference is again performed on each of the individual graphs with this modification to the decomposition, the estimates at the selected node $i$ will all agree, as will the weighted versions of the error variances, $\lambda_k P_i^k$.
- Iterate through these steps, where the selection of the node $i$ should cycle through the nodes of the full graph.
- On convergence, the estimates at *all* nodes in each of the graphs agree and equal the estimate for the original model. Moreover, the weighted error variances, $\lambda_k P_i^k$, will also agree at all nodes and are upper bounds on the corresponding true error variances.

As stated this algorithm requires a set of full inference computations on each of the graphs in our decomposition in each iteration – even though the only change from the preceding iteration is the modification of potentials at a single node. As described in [29], if all of the graphs corresponding to the decomposition (23) are acyclic, we only need to perform a modest part of the full message-passing computation for inference in each subgraph. In particular, if the potential at node $i$ is modified at a particular iteration, and node $j$ will have its potentials modified at the next iteration, we need only perform message-passing to update the messages on the path from node $i$ to node $j$ in each graph in the decomposition. We refer the reader to [30] for a proof that this much more efficient iteration maintains the same convergence properties.

### 2.5. Wavelet-based low-rank variance estimation

A new approach to scalable computation of variances for graphical models, developed in [31,32] is based on the idea of a low-rank approximation to the identity. Let $J$ denote the $N \times N$ information matrix of a graphical model; the covariance $P$, then, is obviously the solution to the equation $JP = I$. Consider replacing the identity matrix $I$ in this equation by $BB^{\mathrm{T}}$, where $B$ is an $N \times M$ matrix, with $M \ll N$ and with rows, $b_i$, $i = 1, \ldots, N$, all of unit norm. Then the diagonal elements of the resulting approximate covariance matrix can be expressed in terms of the true values of the elements of that matrix and the rows of $B$ as follows:

$$\widehat{P}_{ii} \triangleq (J^{-1}(BB^{\mathrm{T}}))_{ii} = P_{ii} + \sum_{i \neq j} P_{ij} b_i^{\mathrm{T}} b_j. \qquad (28)$$

Note that the diagonal elements of $\widehat{P}$ can be computed efficiently, with complexity $\mathcal{O}(MN)$. We first solve the equation $JR = B$ column by column, where the solution for each column corresponds to solving for estimates in a graphical model with the corresponding column of $B$ as $h$. Thanks to the sparsity of $J$ and any of the iterative estimation algorithms described previously (or in the next section), solving each estimation problems has $\mathcal{O}(N)$ complexity. Once we have these solutions we then compute the diagonal elements of $RB^{\mathrm{T}}$.

Note also that according to (28), our approximation to the variance at node $i$ equals the true value plus a set of interference terms. Each of these terms will be small if one of two things is true: either $P_{ij}$ is small – i.e., the variables at nodes $i$ and $j$ are nearly uncorrelated – or the rows $b_i$ and $b_j$ are orthogonal. It is precisely the objective of keeping each of the terms in the summation in (28) small that drives the choice of the rows of $B$ and, in particular, how that choice is determined by the covariance structure of the process being modeled. For example, consider the case of a single-scale model on a grid such as in Fig. 1a and suppose that the MRF in question has primarily local correlations. Then we can imagine choosing a set of rows for, say, the upper left corner of such a field that are standard, orthonormal basis vectors corresponding exactly to the first few rows of an identity matrix. However, when we move a distance away from the upper left corner of this field, we then simply *repeat the same rows* as were used for the upper left corner, with a random flip of sign. We repeat this process, tiling the entire grid with the same set of basis vectors used in the upper left, but with randomly flipped signs. The result is a "spliced basis" as illustrated in Fig. 2a for a 1-D signal.

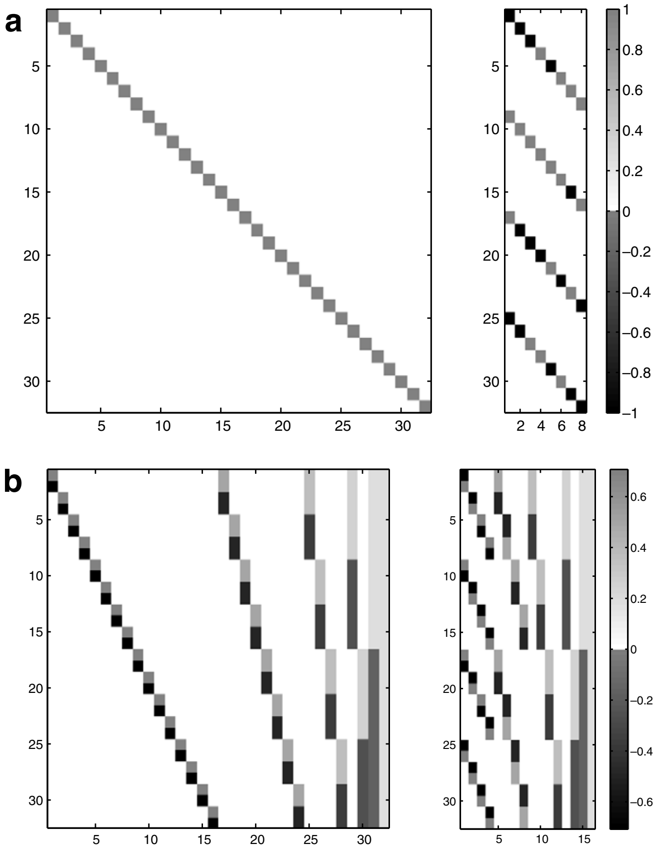


Fig. 2. Illustrating spliced bases for low-rank variance estimation: (a) the full identity matrix and associated spliced basis appropriate for fields with exponentially decaying correlations and (b) a full wavelet basis (illustrated with the Haar basis) and associated spliced basis appropriate for fields with long-distance correlations.

Assuming that we have chosen to start replicating vectors beyond the correlation length of the field, the interference terms on the right-hand side of (28) will be small; moreover, thanks to the random flipping of signs, (28) represents an unbiased estimate of $P_{ii}$, which can be further improved by averaging the results from several computations with different random flips of signs.

For processes with long-distance correlations, this simple approach does not work. However, taking advantage of the well-known property of wavelets, namely that of reducing the correlation length of processes with long-distance correlations, we can, in such cases, use *spliced wavelet bases* such as is illustrated in Fig. 2b. We refer the reader to [31,32] for details and for analysis of accuracy and asymptotic properties. As we develop in Section 3, the structure of our multiresolution models allows us to take advantage of both of the methods (illustrated in Fig. 2a and b) for choosing the matrix $B$ in order to obtain high-quality, unbiased estimates of error variances.

## 3. Multipole stochastic models and algorithms

In this section we introduce our multiscale models, describe some of their qualitative statistical properties, and develop and demonstrate new inference algorithms that exploit both model structure and recent advances in graphical models summarized in Section 2. The models that we introduce here share the motivation that has driven much of the other work on multiscale models – the desire to capture multiresolution characteristics of phenomena and/or data or the objective of computational efficiency in which inference at coarser scales can be used to guide inference at finer scales. In addition, as with some other methods (e.g., see [2–8,21]) the models we describe, based on graphs as in Fig. 1c, overcome the blocky artifacts that result from employing multiscale models defined on trees as in Fig. 1b.

Moreover, we contend that models we introduce here are, in fact, the natural models for many applications. The reason for this stems from the principle of maximum entropy modeling. In particular, for many large-scale data assimilation problems we don't have a complete statistical description of the full field of interest. For example, in describing the statistical variability in the ocean (e.g., variations in sea level that capture large-scale ocean circulation), one might typically have only partial statistical specifications. Specifically, we may have estimates of the variance of sea level variations at different scales and also may have estimates of correlation lengths for different scale features. For example, we may know something about the correlation structure on the order of kilometers near California or near Japan, but we probably do not have reliable information on the correlation of variations at that small scale *between* regions close to California and close to Japan. On the other hand, if we look at sea level variability at coarser scales – e.g., on the order of hundreds of kilometers – we may have estimates of correlations across much larger parts of the ocean.

What this suggests is that the available information we have about such a large-scale process echoes the graphical structure in Fig. 1c (or a slightly more complex version in which there are more edges within each individual scale) – i.e., the statistics we have available might provide correlations among spatial averages of a field at different scales and at spatial separations commensurate with the scale. Of course, this is only a partial specification of the full statistics and might even be inconsistent if the estimated correlations and variances have errors.[10] In either case, the MER methodology can be applied, resulting in a model that matches the specified statistics as accurately as is possible or desired and that has structure as in Fig. 1c.

For these reasons we believe that models with structure as in Fig. 1c and as described next are quite natural multiscale descriptions for many important applications. Of course, there are now significant computational issues to be considered, as Fig. 1c is not a tree and, in fact, has a very large number of loops. Fortunately, the structure and properties of these models allows us to take advantage of the recent advances in inference for loopy models described in the preceding subsections, leading to new, powerful, and insightful algorithms.

### 3.1. The model and its properties

The models we consider are defined on pyramidal graphs as in Fig. 1c. While our framework can readily accommodate data, phenomena, and objectives at multiple scales, we focus our attention here on the case in which our primary interest is in the finest scale in these models – i.e., all measurements and desired estimates are at this scale – so that the role of the coarser scales in the pyramid are simply to aid in inference at the finest scale. Similar objectives can also be gleaned from other, related work – e.g., that involving similar pyramidal graphs in [3,5–7] or ideas from renormalization groups [33]. However, in the former, aggregated versions of finest scale measurements are inserted at coarser scales and dependencies between these aggregate measurements and their fine scale versions are ignored, while in the latter, the complex inter-scale relationships are approximated rather than represented exactly. Of course, if the real objectives are only at the finest scale – so that, in true multigrid fashion [34] the existence of the coarser scales is only to guide and speed up finer scale inference – these approximations need not be material in terms of the ultimate results. In contrast, in our work we begin with a statistically consistent multiscale graphical model, and this allows us to make direct use of emerging methods for graphical models to construct new, powerful, and insightful algorithms.

[10] Testing consistency of a given set of statistics – i.e., the existence of *any* model that matches the specified statistics exactly – is a non-trivial problem. The MER methodology [15] provides a mechanism for relaxing the constraints on specified statistics and producing graphical models that match the available statistics to within specified tolerances.

We denote the coarsest scale[11] in our pyramidal representation as *Scale* 1 and the finest scale as *Scale M*. Let $V_m$ denote the set of nodes at scale $m$, $x_{(m,i)}$ denote the random variable at node $i \in V_m$, $x_m$ denote the vector of all variables at scale $m$, and $x = (x_1^{\mathrm{T}}, x_2^{\mathrm{T}}, \ldots, x_M^{\mathrm{T}})^{\mathrm{T}}$. In addition, for $i \in V_m$, let $N_m(i)$ denote its neighbors at the same scale, and for $m \neq M$, let $C(i)$ denote its neighbors at the next, finer scale. The structure of our Gaussian graphical model, expressed in terms of its information matrix, can be described in terms of two components:

$$J_{\text{prior}} = J_t + J_s, \tag{29}$$

where $J_t$ encodes links between different scales, and $J_s$ represents edges within each scale.

For the sake of specificity in our development we employ generalizations of the so-called thin membrane model for both the intra- and inter-scale information matrices (other choices such as thin-plate models, mixtures of these, etc., are also possible). Thin membrane models capture the idea that each value of the field is most likely to be close in value to its neighbors. However, “neighbors” mean different things within and across-scales, and spatial distances are different at each successive scale. These differences we capture as follows. First, consider $J_t$, which captures the interaction between each parent node and its four children:

$$\exp\{-x^{\mathrm{T}} J_t x\} = \exp\left\{-\sum_{m=1}^{M-1} \beta_m \sum_{i \in V_m} \sum_{j \in C(i)} \left(x_{(m,i)} - x_{(m+1,j)}\right)^2\right\}, \tag{30}$$

where the parameter $\beta_m$ determines how severely we penalize the difference between the value at a node at scale $m$ and the value at each of its children at scale $m+1$. $J_t$ is a block tri-diagonal matrix and can be decomposed by scale as follows:

$$J_t = \begin{pmatrix} c\beta_1 I_{N_1} & \beta_1 J_{T_{12}} & 0 & 0 \\ \beta_1 J_{T_{21}} & (\beta_1 + c\beta_2) I_{N_2} & \beta_2 J_{T_{23}} & 0 \\ 0 & \ddots & \ddots & \ddots \\ 0 & 0 & \beta_{M-1} J_{T_{M,M-1}} & \beta_{M-1} I_{N_M} \end{pmatrix}. \tag{31}$$

Here, $N_m$ is the number of nodes at scale $m$, and $I_{N_m}$ is the $N_m \times N_m$ identity matrix. The constant $c$ is the number of children of each parent, so in our pyramidal graph, $c = 4$. $J_{T_{m,m+1}}$ is a sparse $N_m \times N_{m+1}$ matrix in which each entry corresponding to a parent–child pair equals $-1$, and all other entries are zero. Also, we define $\beta = (\beta_1, \beta_2, \ldots, \beta_{M-1})$.

The nearest-neighbor grid model $J_s$ imposes smoothness within each scale:

$$J_s = \begin{pmatrix} \alpha_1 J_{s1} & 0 & 0 & 0 \\ 0 & \alpha_2 J_{s2} & 0 & 0 \\ & & \ddots & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \alpha_M J_{sM} \end{pmatrix}, \tag{32}$$

where $J_{sm}$ represents a thin membrane prior at scale $m$:

$$\exp\{-x^{\mathrm{T}} J_s x\} = \exp\left\{-\sum_{m=1}^{M} \alpha_m \sum_{i \in V_m} \sum_{j \in N_m(i)} \left(x_{(m,i)} - x_{(m,j)}\right)^2\right\}. \tag{33}$$

Notice that an off-diagonal entry $(J_{sm})_{ij} = -1$ if $j \in N_m(i)$ and 0 otherwise. The diagonal elements of $J_{sm}$ are equal to the number of neighbors each node has within scale $m$. The parameter $\alpha_m$ determines how severely we penalize the gradient of the field at scale $m$. Coarser scale nodes represent spatial regions in which the center points are located farther apart, so it is natural to decrease $\alpha_m$ as we go to a coarser scale. We also define $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_M)$.

Note that the thin membrane model, as well as its extension to a quadtree and multiple grids, yields singular $J$ matrices. Therefore, in order to make $J_{\text{prior}}$ a valid prior model, we add a small regularization term $\epsilon I$ to $J_{\text{prior}}$ to make it positive definite.[12] This small positive diagonal addition can be viewed as adding a weak upper limit on the variance at each node in our model. Further, as long as all of the parameters $\alpha$ and $\beta$ are non-negative, all of the non-zero off-diagonal elements of $J$ are negative, so that all partial correlation coefficients across edges in the graph are positive. Our model is then what is known as an attractive model, the entire class of which is known to be walk-summable [18].

The measurements that we consider are all at the finest scale. In the simplest case these measurements are at individual finest scale nodes, corrupted by independent zero-mean Gaussian noise. However, in some of our examples, our finest scale model represents a discrete set of points on a continuous random field surface. In this case, if a real measurement occurs at a point other than at one of our grid points, we model it as a measurement of a weighted sum of the values at the nearest grid points (using bilinear interpolation). In either case, our measurements take the form $y = Cx + v$, where $v$ is zero-mean with diagonal covariance $S$. As a result, the conditional information matrix for our random field (see (11))

$$J = J_{\text{prior}} + C^{\mathrm{T}} S^{-1} C \tag{34}$$

adds, at worst, some additional local edges at the finest scale. In particular, if a measurement is taken at a point interior to one of the small squares in Fig. 1a, then the effect of bilinear interpolation is to introduce off-diagonal

[11] While Fig. 1c shows this coarsest scale as being a single point, we generally use a coarsest scale that still has spatial extent – the only assumption is that this coarsest scale is tractable, so that exact inference can be performed efficiently.

[12] The regularization term can be made arbitrarily small. In our experiments, we set $\epsilon$ to $10^{-10}$, and the model is not sensitive to the choice of $\epsilon$ as long as it is sufficiently small.

elements in $C^{\mathrm{T}}S^{-1}C$ corresponding to the diagonal edges connecting opposite vertices in that small square.

The class of pyramidal models just introduced can exhibit a range of behaviors depending on the choices of the parameters $\alpha$ and $\beta$. We illustrate here two important qualitative properties that result if we use a particular choice of parameter settings that captures the differing spatial dimensions at different scales. Since the spatial distance[13] between a pair of neighboring nodes at scale $m$ is twice the corresponding distance at scale $m+1$ and since our prior involves the squares of differences, it is appealing to decrease $\alpha_m$ (and $\beta_m$) by a factor of 4 as we move from a finer scale to its parent. Furthermore, since the spatial distance between a child and a parent is $1/\sqrt{2}$ of the distance between a pair of siblings, we are led to set $\beta_{m-1} = \frac{1}{2}\alpha_m$. Therefore, we let $\alpha_M = \varphi$ and set the rest of the parameters as follows:

$$\alpha_m = \frac{\varphi}{4^{M-m}}, \quad m = 1, 2, \ldots, M, \tag{35}$$

$$\beta_m = \frac{1}{2}\frac{\varphi}{4^{M-1-m}}, \quad m = 1, 2, \ldots, M-1. \tag{36}$$

To illustrate the qualitative properties of this model, we use a 1-D process and a total of 4 scales for our multiscale pyramid and tree models (so that the single-scale model is a Markov chain and the tree counterpart is a dyadic tree). We set $\varphi = 1$, which sets the parameters of our pyramidal model. For comparison we use a tree model with the same parameter vector $\beta$ (which is equivalent to setting $\alpha$ to zero in our pyramidal model); for our single-scale model we use the parameter $\alpha_M$ from the pyramidal model. Fig. 3a depicts the correlation decays realized by these three different models. The figure illustrates several important qualitative features. The first is that our pyramidal model displays far stronger long-range correlations than the single-scale counterpart. As a result, this pyramidal model can more easily capture such long-range correlations – e.g., as found in processes with $1/f$ – behavior. To be sure, single-scale models can do this as well, but such models will of necessity be of much higher-order (corresponding, for example, to far more connected single-scale graphs). As is well known, tree models can also capture long-distance correlations, but their limited modeling capability leads to the blocky, staircase correlation structure present in the figure. Of course more complex, higher-order tree models (e.g., those involving so-called overlapping trees or keeping vectors of wavelet coefficients at each node [35,36]) can be used to smooth out these artifacts, but our pyramidal models do this without such added complexity.

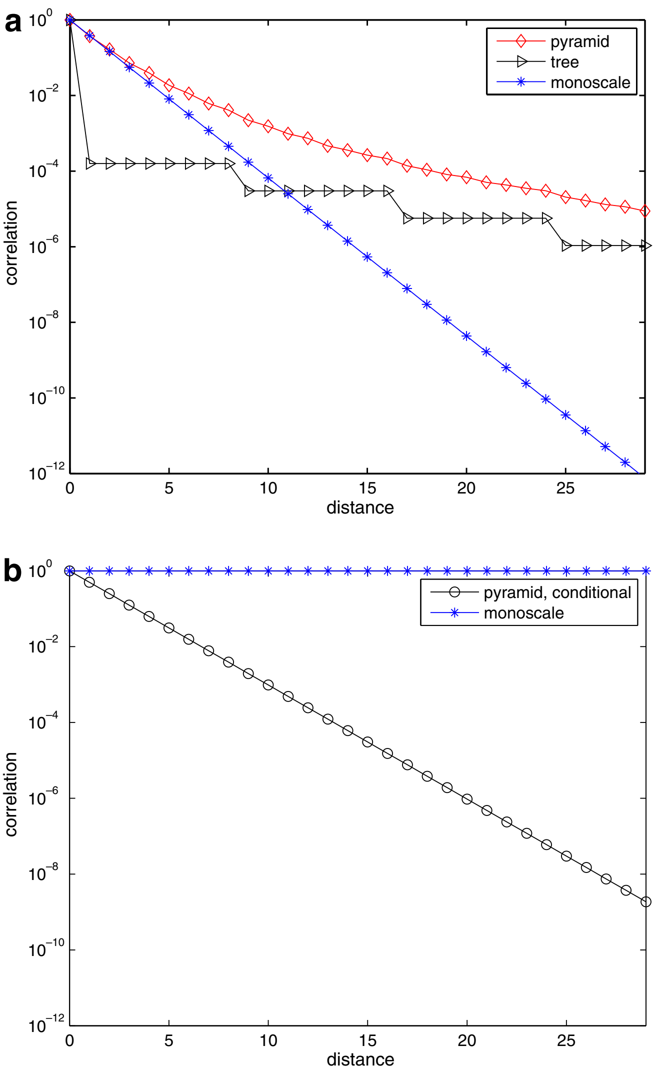


Fig. 3. The correlation decays realized by a pyramidal graph, a tree, and a single-scale model: (a) correlations conditioned on measurements. For the pyramidal graph and the tree, correlations at the finest scales are plotted and (b) correlations at the finest scale of the pyramidal graph conditioned on coarser scales and correlations of its single-scale counterpart.

Fig. 3b illustrates a second very powerful property of our pyramidal models – and one that has significant implications for our inference algorithms. Specifically, in this figure we show the *conditional correlation* at the finest scale when we condition on the coarser scales (rather than on measurements). Note that this conditional correlation shows substantial, exponential decay. Indeed, as shown in [30] the ratio of the condition number of the covariance matrix at the finest scale for our pyramidal model and the condition number for the covariance at the finest scale conditioned on coarser scales is enormous (17 orders of magnitude for the parameter settings used in our figures). This suggest not only that coarse-to-fine algorithmic structures will offer considerable advantages but also that each fine scale iteration – i.e., the computation of conditional walk-sums at each scale – can be well approximated by only a few iterations (i.e., by very short walks), as the residual correlation that needs to be captured is concentrated very locally. This is reminiscent of the structure of so-called *multipole* algorithms [37] in which corrections to coarser scale approximations are implemented using FIR filters. As we will see, our use of the emerging graphical model

[13] Each node in the pyramidal graph represents a region on a plane. The spatial distance between two nodes is the distance between the center points of the corresponding regions.

methods described in Section 2 leads to algorithms that exploit these qualitative properties.

### 3.2. Computation of estimates

In this section, we describe two new approaches to performing estimation for the class of models described in the preceding section. These methods make extensive use of the recent advances in inference for graphical models summarized in Section 2. Our first approach is reminiscent of multipole algorithms for the solution of elliptic PDEs, namely an approach in which we use coarser scale estimates to capture longer-scale effects, which then allows us to take advantage of the much more local conditional correlation within each scale. Embedding these ideas within the framework of graphical inference leads to embedded subgraph algorithms with structure inherited from our pyramidal models and with guaranteed convergence, thanks to walk-summability. Moreover, using the methods summarized in Section 2.4 and [23], we can choose the embedded subgraphs adaptively in each iteration, accelerating convergence considerably.

Since we can view our pyramidal model as a 1-D Markov chain in-scale, there is a conceptually simple two-sweep optimal estimation algorithm: we propagate messages from fine-to-coarse first,[14] and then conclude with a coarse-to-fine sweep. This approach requires that exact inference be performed at each scale in both sweeps, as this is needed to compute both the estimates at each scale as well as the "messages" passed to adjacent scales. While one can certainly imagine doing this using the iterative methods described in Section 2 for the operations within each scale, this naive approach collects "walks" in a manner that does not make maximal use of the structure of our models. To take advantage of this requires that we be more judicious in which computations we perform in each fine-to-coarse and coarse-to-fine step, thereby making each step in these sweeps far simpler but, then also requiring their iterative application (to capture walks not included in each simplified step). We first describe how this is done with a fixed structure for the subgraphs used in each step of each sweep of the algorithm.

Our algorithm first performs a coarse-to-fine sweep using the natural embedded quadtree of Fig. 1b – i.e., we compute the upward, Gaussian elimination step using only the tree-based component, $J_t$, of our prior model (29) and ignoring $J_s$ – with one exception. Specifically, note that if we use a full pyramid as in Fig. 1c, the coarsest scale consists of a single node so that there is no contribution from $J_s$ at that coarsest scale. More generally, if we truncate our pyramid at a resolution so that the coarsest scale itself has a non-trivial grid, we assume that it is still small enough so that exact inference at this scale is tractable. Consequently, this upward sweep performs coarse-to-fine computations exactly as in quadtree models, with the understanding that at the coarsest scale, this computation involves exact computations on a small grid of variables.

This initial upward sweep produces estimates at each scale, and we then perform a coarse-to-fine sweep, followed by alternating sweeps from fine to coarse and coarse-to-fine. In each step of the coarse-to-fine sweep we update the estimates at one scale, say $m$, using the just-computed estimates at its coarser neighbor, $m-1$, and the previous upward sweep estimates at the next finer scale, $m+1$. A complete version of this update corresponds to solving the equations:

$$J_{[m,m]}\hat{x}^{\mathrm{d}}_m = h_m - J_{[m,m-1]}\hat{x}^{\mathrm{d}}_{m-1} - J_{[m,m+1]}\hat{x}^{\mathrm{u}}_{m+1}, \tag{37}$$

where $J_{[j,k]}$ denotes the $(j,k)$-block of the full $J$ matrix in (34), and the superscripts "d" and "u" are included to emphasize the structure of the coarse-to-fine step, namely in using the just-computed "downward" sweep estimates at scale $m-1$ and the previously computed "upward" sweep estimates at scale $m+1$. Also, $h_m = 0$ for all scales other than $m = M$ (since we only have data at the finest scale), and $\hat{x}^{\mathrm{u}}_{M+1} \equiv 0$. This recursion begins at scale 2, using the just-computed coarsest scale estimates from the preceding fine-to-coarse iteration. However, at each of these scales, solving (37) is typically intractable, thanks to the sizes of the grids involved. Consequently, rather than solving this equation exactly, we can instead perform several ET iterations within scale – i.e., we decompose $J_{[m,m]}$ into one term on an acyclic graph and a "cutting matrix" term, i.e., we rewrite (37) as[15]

$$J_a\hat{x}^{\mathrm{d}}_m = h_m - K_a\hat{x}^{\mathrm{u}}_m - J_{[m,m-1]}\hat{x}^{\mathrm{d}}_{m-1} - J_{[m,m+1]}\hat{x}^{\mathrm{u}}_{m+1}, \tag{38}$$

where $J_{[m,m]} = J_a + K_a$, and perform one or more iterations using this decomposition, as suggested in (18). Thanks to the multipole character of our models, we expect that the additional smoothing required conditioned on the coarser scale, is local in nature. A decomposition that makes this completely evident is simply to cut all edges, i.e., to take $J_a$ to be diagonal.

Each subsequent fine-to-coarse sweep uses the just-computed estimates from the downward sweep and performs the fine-to-coarse portion of an ET iteration (see (18)) using the quadtree in Fig. 1b with all in-scale edges cut. The resulting overall algorithm thus involves a mixture of global, fine-to-coarse ET iterations and scale-by-scale local computations within each scale during each coarse-to-fine sweep, in which the computation at each scale is itself replaced by an ET iteration. Thanks to walk-summability this algorithm is guaranteed to converge to the optimal estimates. Moreover, we can make this algorithm adaptive, and hence speed up convergence, by adaptively choosing the spanning trees used at each step of the process. Specifically, rather than constraining ourselves to using a completely disconnected

[14] Note that since we are assuming that data are available only at the finest scale – i.e., the only non-zero elements of $h$ are at the finest scale, it only makes sense to start with a fine-to-coarse sweep.

[15] Notice that $\hat{x}_m$ on the right-hand side has the superscript $u$, indicating that the *previous* upward sweep estimates at scale $m$ is used to compute the term $K_a\hat{x}^{\mathrm{u}}_m$.

Table 1
Multipole-motivated inference algorithm using the adaptive ET iterations

(1) *Initialization:* get initial estimates based on the tree prior and the thin membrane prior within the coarsest scale
(2) *In-scale inference:* starting from the coarsest scale and proceeding to finer scales, smooth the estimates using an adaptively chosen spanning tree within each scale
(3) *Tree-inference:* apply one ET iteration using a spanning tree embedded in the pyramid chosen by the adaptive ET algorithm
(4) Repeat the in-scale inference and tree-inference steps until a stopping criterion is met

tree for each in-scale iteration, we can choose a max-weight spanning tree to accomplish this using the method summarized in Section 2.4. Similarly, in the upward ET sweep, rather than using the standard quadtree as in Fig. 1b, we can use the same max-weight spanning tree algorithm to choose an embedded spanning tree for each fine-to-coarse iteration over the entire pyramidal graph. The steps in this adaptive algorithm are summarized in Table 1.[16]

Fig. 4 depicts several experiments on a 64 × 64 synthetic example, with the true field to be reconstructed given in Fig. 4a, two dense sets of measurements with low-level noise (Fig. 4b, with a per-pixel SNR of approximately 20), high-level noise (Fig. 4c with a per-pixel SNR of 6), and sparse measurements with low-level noise in Fig. 4d. Fig. 5 depicts the resulting estimates for these three cases, and Fig. 6 compares the convergence performance of our pyramidal multipole algorithm (referred to as "pyramid" in the figure) with two different algorithms based on a single-scale thin membrane model. One of these (referred to as "monoscale" in the figure) uses a Gauss–Jacobi iteration directly on this model. The other uses a standard multigrid algorithm for solving such single-scale problems efficiently in which approximate, coarser versions of the problem are solved in order to guide the finer version. The plots on the left in Fig. 6 are for algorithms with fixed subgraphs used in each iteration (Gauss–Jacobi within each scale), while the plots on the right are for adaptive ET iterations described in Table 1.

Since we know truth for this example, the error measure used in these plots compares estimates to truth. In particular at iteration $n$ we compute the RMS error as

$$e^{(n)}_{\text{rms}} = \sqrt{\frac{\sum_{i\in V_M}(x_i - \hat{x}^{(n)}_i)^2}{N_M}} \tag{39}$$

where $\hat{x}^{(n)}$ denotes the vector of estimates at iteration $n$ and $N_M$ is the number of nodes at the finest scale. As a full iteration in either our pyramidal multipole algorithm or in the multigrid algorithm involve more computations than a single monoscale Gauss–Jacobi step, the horizontal axes in Fig. 6 are in units of "equivalent" monoscale iterations. As one would expect, the monoscale algorithm converges much more slowly than our pyramidal model which, after a very few iterations using either fixed (left side of Fig. 6) or adaptive (right side) subgraphs,[17] achieves performance comparable to multigrid. This would seem to imply that the method to prefer is the multigrid approach, and indeed, if one were only interested in these estimates, that might very well be the case. However, we may be interested in much more – e.g., computing error variances, in solving efficiently what we refer to as the re-estimation problem, or in computing estimates of parameters – and in such situations, it is important to have consistent statistical models, which is precisely what our pyramidal models provide.

The multipole algorithm just described provides us with optimal estimates but not variances. In the next subsection we will have more to say about computing good approximations to the error variances, but for now we turn to a second approach, namely that based on Lagrangian relaxation, which also provides bounds on variances. We apply the general ideas introduced in Section 2.4 by decomposing our graphical model into a set of models on subgraphs that separate the multiscale and within scale structure. One of the graphs used in our decomposition is that capturing only the multiscale interactions, i.e., the quadtree in Fig. 1b. While one choice for the remaining subgraphs is the set of 2-D grids at each individual scale, inference on these graphs themselves is complex. While we can overcome this by performing inference within each scale using iterative procedures (such as an ET-based procedure), we choose here to illustrate the Lagrangian relaxation methodology by further decomposing the grid within each scale into two spanning subgraphs, one including all horizontal edges within the scale and the other including all vertical edges. Note that with this full decomposition, each edge of the pyramidal model in Fig. 1c appears in only one of the component subgraphs. Note also that while one of these subgraphs is the quadtree that is also used in the multipole algorithm described previously, its use here is substantially different, as the key idea in Lagrangian relaxation is that of ensuring that the (weighted) inference results on the quadtree and on each of the individual in-scale subgraphs agree.

Fig. 7 depicts the result of applying Lagrangian relaxation to the estimation of the field in Fig. 4a based on the sparse measurements in Fig. 4d. Here the initial decomposition used in the Lagrangian relaxation is as follows:

$$\begin{aligned} J^1 &= J_t + (1-2\delta)C^{\mathrm{T}}S^{-1}C \quad h^1 = (1-2\delta)C^{\mathrm{T}}S^{-1}y,\\ J^2 &= J_{sv} + \delta C^{\mathrm{T}}S^{-1}C \quad h^2 = \delta C^{\mathrm{T}}S^{-1}y,\\ J^3 &= J_{sh} + \delta C^{\mathrm{T}}S^{-1}C \quad h^3 = \delta C^{\mathrm{T}}S^{-1}y, \end{aligned} \tag{40}$$

where $J_{sv}$ and $J_{sh}$ denote the vertical and horizontal portions of the smoothness terms within each scale captured together in $J_s$ in (32). The constant $\delta$ assigns weights of

[16] For the non-adaptive version of this algorithm, the in-scale coarse-to-fine Step 2 uses the completely disconnected graph, which reduces the in-scale computations to several Gauss–Jacobi iterations, and the standard quadtree in Fig. 1b is used in the fine-to-coarse Step 3.

[17] While for this example there is little difference between the performance using fixed or adaptive subgraphs, there are cases in which using adaptively chosen subgraphs can offer speed ups. We refer the reader to the next subsection on re-estimation problems.

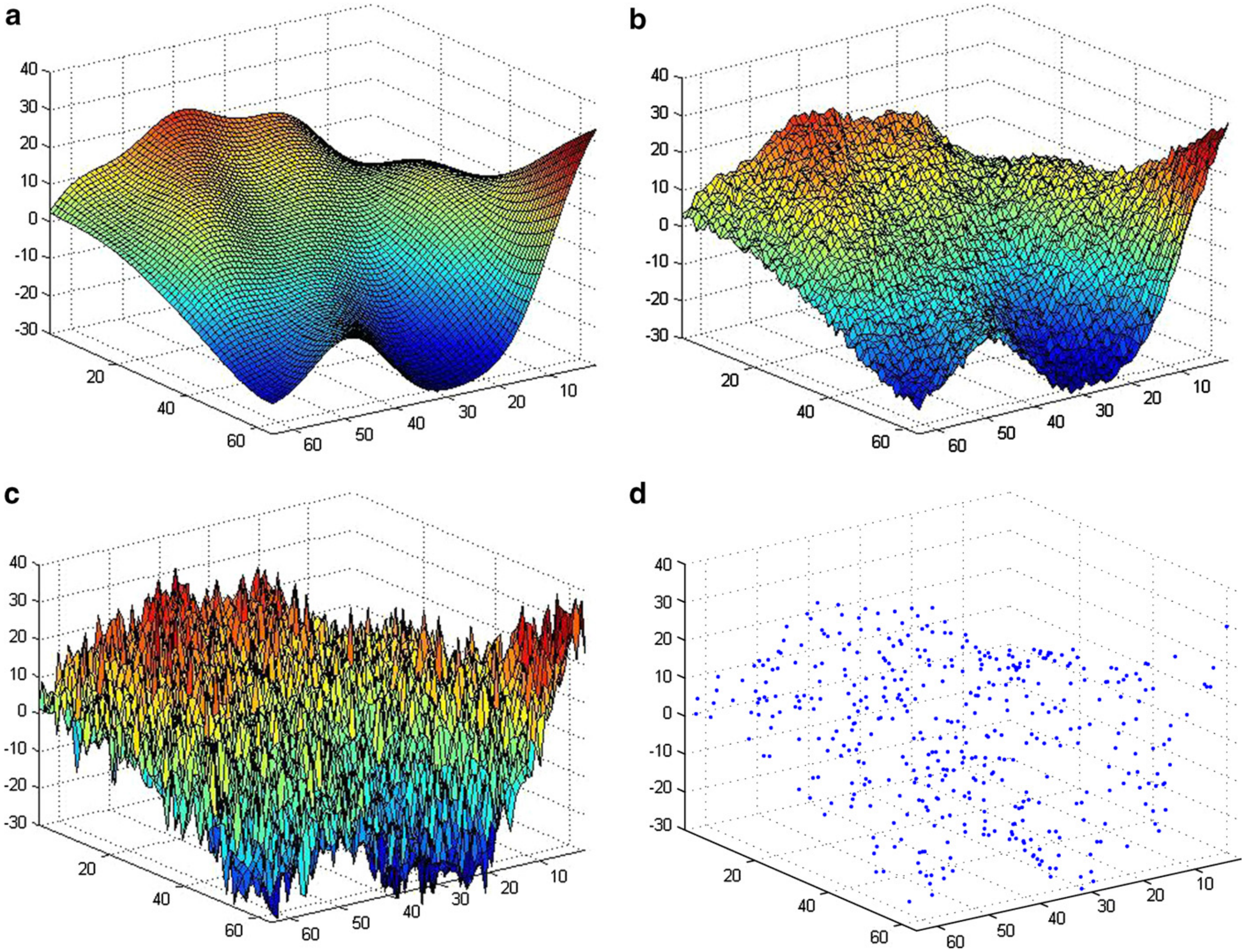


Fig. 4. Test surface and measurements: (a) true surface; (b) dense measurements with low-level noise ($\sigma^2$=1); (c) dense measurements with high-level noise ($\sigma^2$=25) and (d) sparse measurements (10% of the finest scale nodes) with low-level noise ($\sigma^2$=1).

measurements to each subgraph and can be taken as any value between 0 and 1 to ensure that the $J^k$ are positive definite. For the example shown here we set $\delta = 1/3$, which corresponds to the equal distribution of measurements to each subgraph. Note in Fig. 7 that while initially the estimates produced using these different subgraphs yield different estimates (with the expected artifacts), at convergence, these estimates agree and the artifacts are no longer present.

In our experiments the Lagrangian relaxation method required more computational effort than the multipole algorithm (although both of them scale in the same manner with problem size). Of course the Lagrangian relaxation algorithms also gives bounds on error variances at the same time and thus might be better compared to the combined use of our multipole algorithm for the computation of estimates and the method for computing unbiased estimates of variances described subsequently in Section 3.4. The tradeoff in that comparison – using Lagrangian relaxation for both estimates and bounds on variances versus the use of multipole estimation and unbiased variance estimation – is more complex, as it involves a tradeoff of computation (although both variations scale similarly with problem size) versus accuracy (as the Lagrangian relaxation method provides only bounds on variances rather than unbiased estimates).[18]

### 3.3. Re-estimation

An interesting and important variation on the estimation problem on which we have focused is what we refer to as the *Re-Estimation Problem*. In particular, to this point in our development we have assumed that both the prior model and the available data are provided to us at the start and all that is desired are estimates based on these specifications. However, there are two related but distinct reasons to examine problems in which, we wish to change the prior model or provide additional data after estimates based on our previous specification have been computed. The more obvious of these is the case in which we find that, based on the originally supplied data, the smoothness prior, both

[18] Another advantage of Lagrangian relaxation method is that it is easily extendible to non-Gaussian problems.

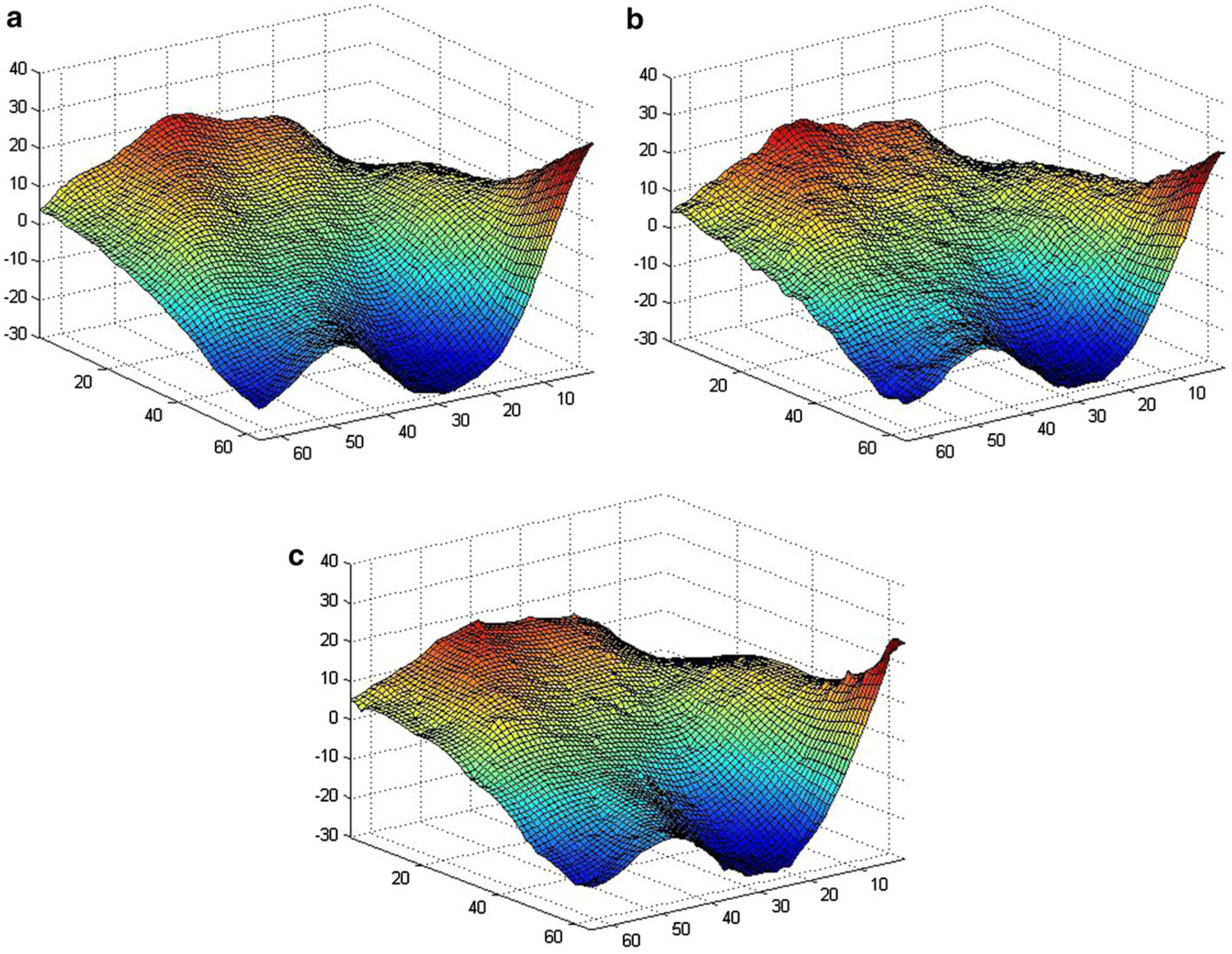


Fig. 5. Estimates using adaptive ET iterations on the pyramidal graph when the normalized residual is reduced to 0.01: (a) dense measurements with low-level noise; (b) dense measurements with high-level noise; and (c) sparse measurements with low-level noise.

within and across-scales, is not appropriate in some regions of the field being estimated, e.g., because of the presence of abrupt discontinuities or cliffs. Such cliffs might be detected automatically by pinpointing locations at which measurements and estimates differ significantly compared to the variation consistent with the variance computed at that location.[19] An alternative – one that we have encountered in developing interactive algorithms with analysts – involves external specification, e.g., by human expert, of changes that must be made in the prior model (such as excluding smoothness terms either within scale or across-scale at particular locations or in particular regions).

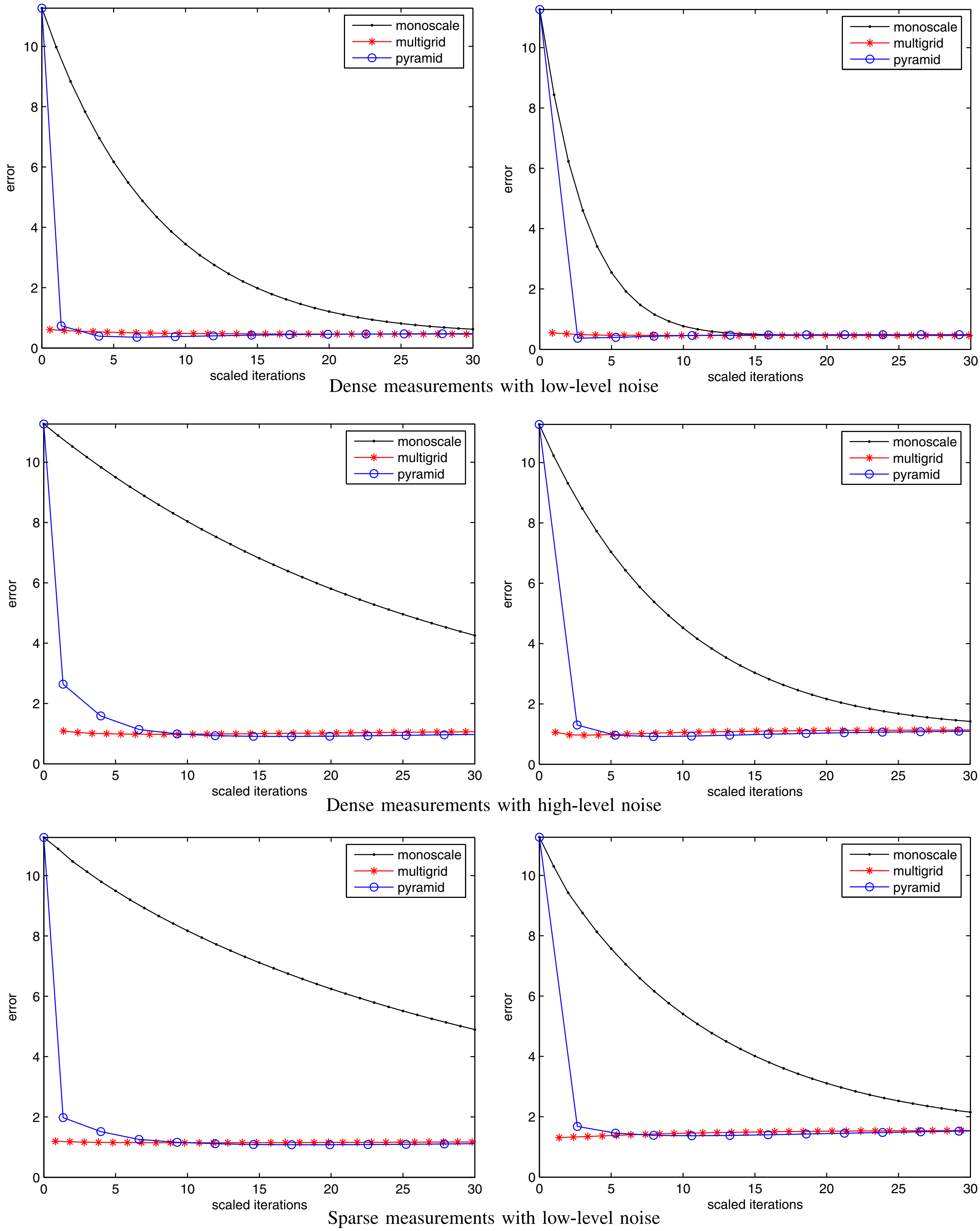
There are also cases in which we wish to update estimates given some additional measurement points. This can occur in some situations in which such data become available over time (e.g., as may be the case in remote sensing applications) or in situations in which an analyst, unsatisfied with the current estimate, injects additional measurements in order to improve the resulting estimate. In either of these cases, the problem of re-estimation can be stated simply as follows:

- *Re-estimation problem:* Suppose that we have $\hat{x} = J^{-1}h$. Efficiently compute the updated estimates $\tilde{x} = (J + \Delta J)^{-1}(h + \Delta h)$, where $\Delta J$ and $\Delta h$ have non-zero elements only in a localized area.

Here a "localized" area might correspond to a small set of nearby points or an entire ridge or narrow swath of points corresponding to a detected discontinuity. The associated changes $\Delta J$ and $\Delta h$ correspond to changes in the prior model, both in- and across-scale and the addition of new measurements.[20]

[19] Of course such an automatic detection method requires that we be able to compute the variances across the field, a topic considered a bit further in the following subsection and also in [31,32]. Also, see [36] for an example of such an approach for multiresolution models on quadtrees.

[20] Note that from (10), (11) changes in the prior model induce changes only in $J$, while the incorporation of additional measurements induces changes in both $J$ and $h$.



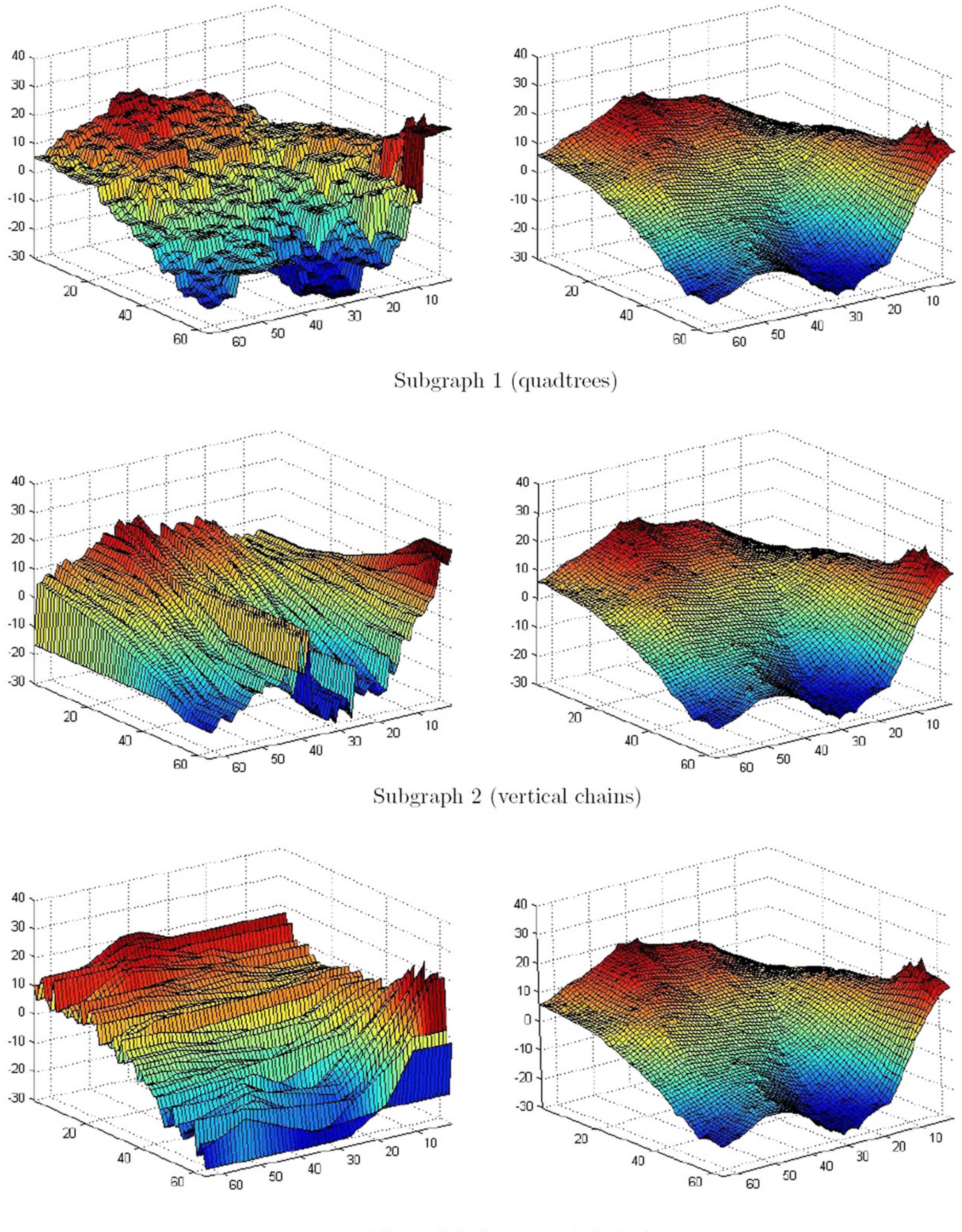
Dense measurements with low-level noise

Dense measurements with high-level noise

Sparse measurements with low-level noise

Fig. 6. RMS errors in surface estimation using multipole-motivated algorithms on the pyramidal graph and corresponding multigrid methods and iterations on the monoscale model. Left: fixed subgraphs iterations (Gauss–Jacobi). Right: adaptive ET iterations.

Of course if we initialize any of the iterative algorithms we have described with the estimates prior to these changes and carry out iterations with the changes, we will eventually converge to the new optimal estimates. However, what we seek



Subgraph 1 (quadtrees)

Subgraph 2 (vertical chains)

Subgraph 3 (horizontal chains)

Fig. 7. Estimates using the Lagrangian relaxation method for sparse measurements. Left: Estimates after the initialization step. Right: Estimates after convergence.

are algorithms that can do this most rapidly. Because of the locality of these changes and the multipole statistical structure of our models, we expect that changes will be at finer scales only near the regions where changes have been made, and that primarily coarser scale changes need to be made in regions farther away. This suggests an adaptive structure with some similarities to the multipole algorithm described previously and summarized in Table 1 but with steps adapted to

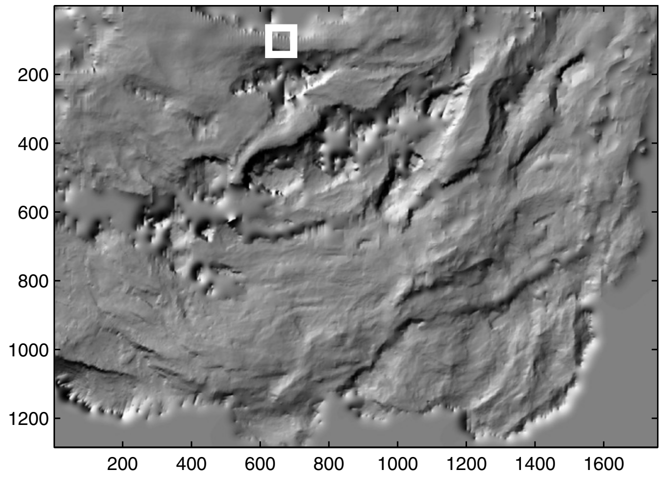


Fig. 8. The estimates of a complex surface from operator-supplied sparse data.

the locality of the changes that require re-estimation. Specifically, let $\mathscr{S}$ denote the region at the finest scale in which changes have been made, i.e., in which either $\Delta J$ or $\Delta h$ is non-zero. Also, let $\mathscr{T}_{\mathscr{S}}$ denote the set of (disjoint) quadtrees, each of which is rooted at a single node at the coarsest scale and which has finest scale nodes that have non-empty intersection with the nodes in $\mathscr{S}$. Our algorithm alternates between tree-based inference iterations in parallel on these quadtrees and adaptive Gauss–Seidel iterations using the adaptive algorithm described in [23] and summarized in Section 2.4 in order to choose a subset of variables to be updated. The latter steps provide rapid estimate adjustments, primarily at finer scales and in the vicinity of $\mathscr{S}$, while the tree-inference steps propagate these estimates more broadly across the field.

Fig. 8 depicts an example of surface reconstruction from human-operator input data points. This is typically an iterative process in which the operator will examine a reconstruction, pinpoint regions in which the smoothness penalties need to be relaxed or removed, and also add some additional data points. Because of the human interactive nature of such a process, it is essential that re-estimation be performed rapidly. The 1757 × 1284 surface shown in Fig. 8 is the result of applying our pyramidal estimation algorithm (using 4 scales) to a sparse set of 377,384 human- and computer entered measurements scattered throughout the region. The total number of nodes in the pyramidal graph is on the order of 3 million. We introduce an additional 100 measurements in the small 17 × 17 square indicated in the figure and apply our re-estimation algorithm, which uses tree-inference steps involving 765 nodes and adaptive Gauss–Seidel steps using 100 nodes. Fig. 9 depicts a zoomed-in look at the updated estimates at convergence (achieved after only 10 iterations) as well as a cross-section comparing the original estimates, those resulting form our re-estimation algorithm and truly optimal re-estimates obtained by re-solving the estimation problem. Note that the re-estimates and the new optimal estimates are indistinguishable; however, the completely recomputed estimates required iterative computation at all 3 million nodes of the pyramid, while the re-estimation algorithm adjusted fewer than 1000 nodes in each of its 10 iterations.

### 3.4. Computation of variances

In this section, we describe two different approximate methods for the scalable computation of approximate error variances for our pyramidal model. The first begins with an exact expression for the marginal covariance at each scale obtained by viewing our multiscale model as a Markov chain in-scale. As derived in [30], the error covariance at scale $m$ is given by

$$P_{[m,m]} = (J_{[m,m]})^{-1} + (J_{[m,m]})^{-1} Q_m (J_{[m,m]})^{-1}, \qquad (41)$$

where

$$Q_m = \begin{cases} \begin{pmatrix} J_{[m-1,m]} \\ J_{[m+1,m]} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} P_{[m-1,m-1]} & P_{[m-1,m+1]} \\ P_{[m+1,m-1]} & P_{[m+1,m+1]} \end{pmatrix} \begin{pmatrix} J_{[m-1,m]} \\ J_{[m+1,m]} \end{pmatrix} & 1 < m < M, \\ J_{[1,2]} P_{[2,2]} J_{[2,1]} & m = 1, \\ J_{[M,M-1]} P_{[M-1,M-1]} J_{[M-1,M]} & m = M. \end{cases} \qquad (42)$$

Here $\overline{P}_{[m,m]} \triangleq (J_{[m,m]})^{-1}$ is the conditional covariance at scale $m$ conditioned on its parent scale $m-1$ and its child scale $m+1$. We denote the $(i,j)$ entry of $\overline{P}_{[m,m]}$ as $\bar{p}_{ij}$. Then, we have the following lower bound on the error variance at node $i$:

$$\begin{aligned} p_{ii} &= \bar{p}_{ii} + \sum_{j \in V_m} \sum_{k \in V_m} \bar{p}_{ij} \cdot \bar{p}_{ik} \cdot (Q_m)_{jk} \\ &> \bar{p}_{ii} + \bar{p}_{ii} \cdot \bar{p}_{ii} \cdot (Q_m)_{ii} + \sum_{j \in N_m(i)} \sum_{k \in N_m(i)} \bar{p}_{ij} \cdot \bar{p}_{ik} \cdot (Q_m)_{jk}, \end{aligned} \qquad (43)$$

where $N_m(i)$ is the set of neighboring nodes of $i$ within scale $m$. The first equality follows from (41), and the inequality follows from the fact that within the pyramidal graph, every partial correlation coefficient and hence every walk-sum is positive. Then the covariance, as well as the conditional covariance between any pair of nodes is positive. It is also straightforward to show that every element of the matrix $Q_m$ is also non-negative.

As we have seen, conditional covariances decay quickly, so $\bar{p}_{ij}$ becomes very small when $i$ and $j$ are not neighbors. Therefore, although the lower bound in (43) is not tight,

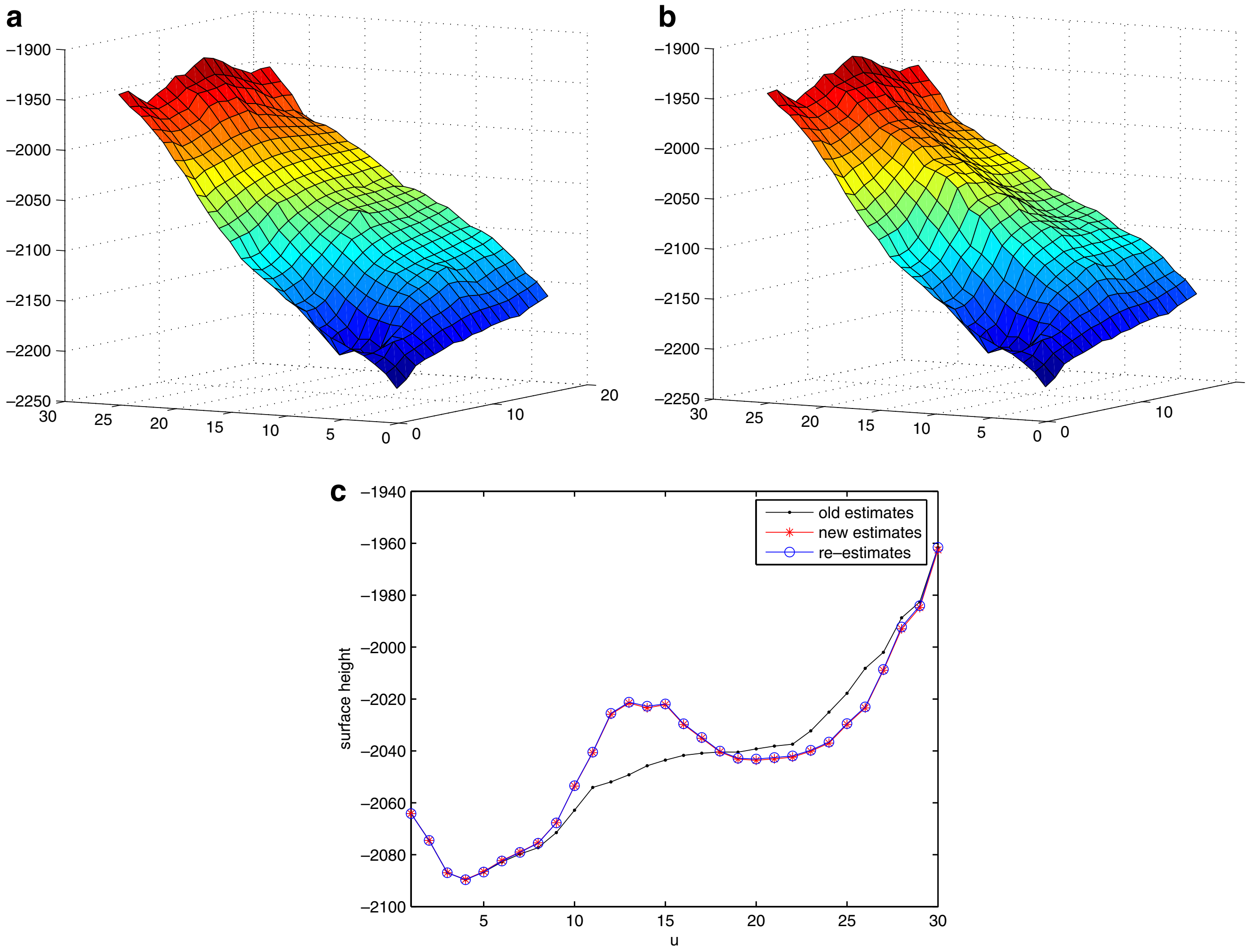


Fig. 9. Re-estimation applied to the problem of updating estimates to incorporate a new set of measurements in a local region: (a) estimates before adding measurements; (b) re-estimates; and (c) a cross-section of re-estimates.

it closely approximates the true value. In addition, we can estimate $\bar{p}_{ij}$ rapidly for $j = i$ or $j \in N_m(i)$ using the low-rank variance approximation algorithm [31] discussed in Section 2.5.

This is only part of the story, as we must also consider the computation of the matrix $Q_m$ in (42). While the various blocks of the $J$ matrix appearing in this expression are sparse (with no more than four non-zero elements per row), the middle term in the factor is in general full, representing the joint covariance of all nodes at scales $m-1$ and $m+1$. Exact computation of this matrix scales cubically with the number of nodes at these scales, and its storage scales quadratically. This is problematic (especially at finer scales) and, in fact, becomes infeasible for large random fields. As the entire idea of using multiscale models such as those developed here is to develop completely scalable methods, we are led to introduce an approximation. In particular, since we are interested in computing only a subset of the elements of $Q_m$, we further relax the bound in (43) and replace this joint covariance for $m-1$ and $m+1$ by keeping only the diagonal elements. This then leads to a coupled set of equations for approximate variances which can be iterated using coarse-to-fine sweeps. Let $\rho_i^{(n)}$ denote the approximate variance at node $i \in V_m$ computed in the $n$th iteration. Then, from (43) we have that

Table 2
The coarse-to-fine variance computation using the low-rank approximation algorithm

```
(1) Compute the exact variances of nodes at scale 1
(2) For all finer scales, use the low-rank variance approximation
    algorithm (see Section 2.5) to compute conditional covariance
    $\overline{P}_{[m,m]}$ conditioned on adjacent scales
(3) Initialize the variances of nodes at finer scales as the conditional
    variances computed at Step 2, i.e. $\rho_i^{(0)} = \bar{p}_{ii}$ for all $i \in V \setminus V_1$
(4) At $n$th coarse-to-fine sweep, for $m = 2, 3, \ldots, M$:
    (a) Compute $\widetilde{Q}_m^{(n)}$ in (45) using the approximate variances of
        nodes at adjacent scales
    (b) Compute the lower bound on variances $\rho_i^{(n)} < p_{ii}$ for $i \in V_m$
        using (44)
(5) Repeat Step 4 until a stopping criterion is met
```

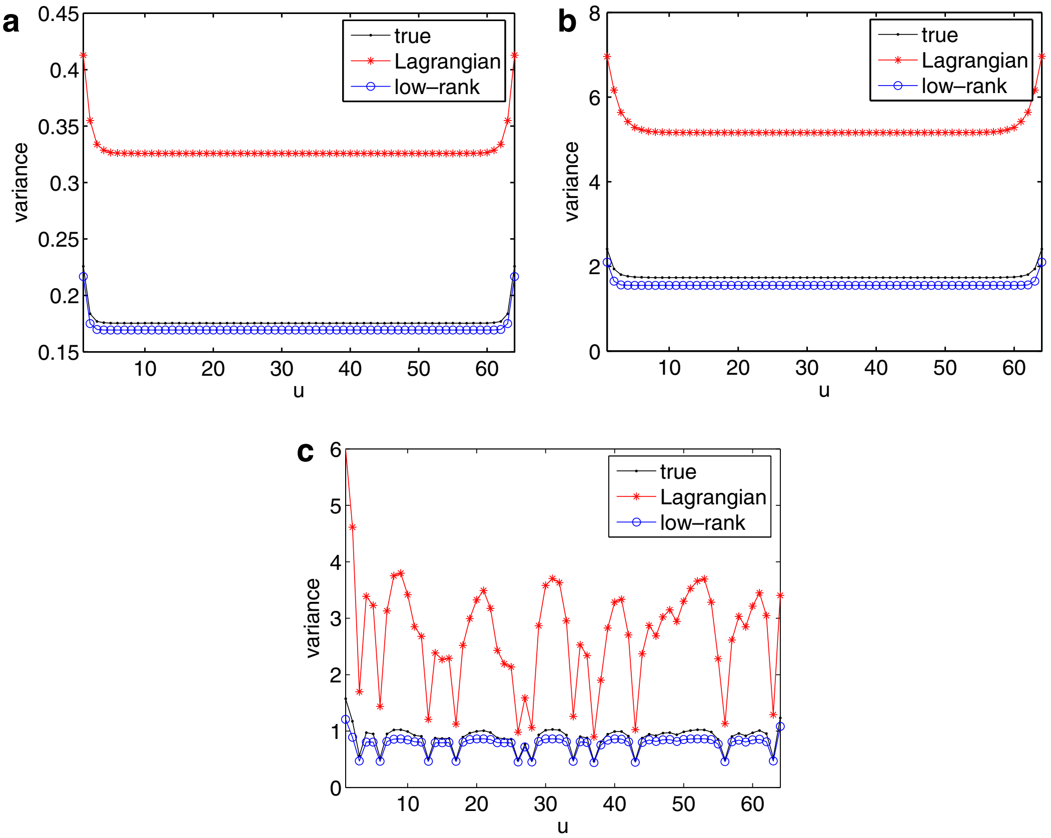


Fig. 10. A cross-section of estimates of approximate variances using the Lagrangian relaxation methods and the coarse-to-fine low-rank algorithm on the pyramidal graph: (a) dense measurements with low-level noise; (b) dense measurements with high-level noise; and (c) sparse measurements with low-level noise.

$$\rho_i^{(n)} = \bar{p}_{ii} + \bar{p}_{ii}\cdot\bar{p}_{ii}\cdot(\widetilde{Q}_m^{(n)})_{ii} + \sum_{j\in N_m(i)}\sum_{k\in N_m(i)}\bar{p}_{ij}\cdot\bar{p}_{ik} \cdot(\widetilde{Q}_m^{(n)})_{jk}, \tag{44}$$

where $\widetilde{Q}_m^{(n)}$ is defined as follows:

$$\widetilde{Q}_m^{(n)} = \begin{cases} \begin{pmatrix} J_{[m-1,m]} \\ J_{[m+1,m]} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \Upsilon_{[m-1]}^{(n)} & 0 \\ 0 & \Upsilon_{[m+1]}^{(n-1)} \end{pmatrix} \begin{pmatrix} J_{[m-1,m]} \\ J_{[m+1,m]} \end{pmatrix} & 1 < m < M, \\ J_{[M,M-1]}\Upsilon_{[M-1]}^{(n)}J_{[M-1,M]} & m = M, \end{cases} \tag{45}$$

$\Upsilon_{[m]}^{(n)}$ is a diagonal matrix with $i$th diagonal element corresponding to the approximate variance at $i \in V_m$ computed at $n$th coarse-to-fine sweep, i.e., $\rho_i^{(n)}$. It is tractable to compute the variances of the nodes at the coarsest scale exactly, so we define $\Upsilon_{[1]}^{(n)}$ to be a diagonal matrix with entries taken from $P_{[1,1]}$. Table 2 summarizes the iterative variance approximation algorithm.

Fig. 10 illustrates this bound (referred to as "low-rank" in the figure) as well as the upper bound provided directly by Lagrangian relaxation for the example shown in Fig. 4. Note that the Lagrangian relaxation bound is quite loose, while in these cases the lower bounds are quite close to the true values. This is not, however, always the case, as in some sparse measurement cases even conditional correlations decay slowly. An alternative that provides accurate variances even in such cases is the low-rank method using spliced wavelet bases developed in [32]. In our case, if we are only interested in variances at the finest scale, we can take $B = (0, B_M^{\mathrm{T}})^{\mathrm{T}}$ with 0 for all coarser scales. Fig. 11 illustrates the very accurate variances estimates obtained for the same three examples using this method.

### 3.5. Parameter estimation

The tractable methods we have described both for the computation of estimates and variances allows us to derive an efficient expectation-maximization (EM) algorithm [2,38] for the estimation of the parameters of our model, namely the parameter $\varphi$ that controls the within- and across-scale smoothness terms in our prior model (see (35) and (36)) and $\gamma$, the reciprocal of the measurement noise variance. If we define the parameter vector $\theta = (\theta_1, \theta_2) = (\varphi, \gamma)$, then our parameterized probability distribution can be written as in (2), with $\Phi(\theta) = \Phi(\theta_1) + \Phi(\theta_2)$, with[21]

[21] Here we have ignored the constant $\frac{1}{2}N\log(2\pi)$ in the log-partition function.

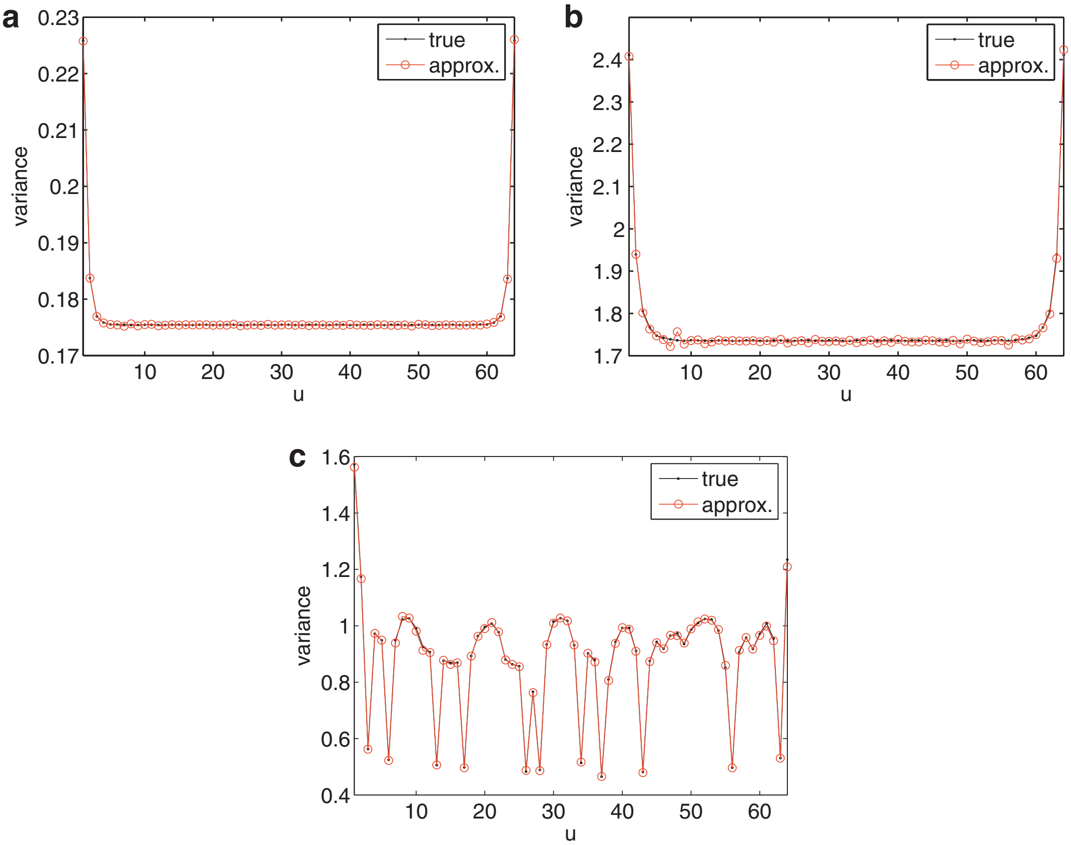


Fig. 11. A cross-section of estimates of variances using the wavelet-based low-rank approach on the pyramidal graph: (a) dense measurements with low-level noise; (b) dense measurements with high-level noise; and (c) sparse measurements with low-level noise.

$$
\begin{aligned}
\phi_1(x) &= -\frac{1}{2}x^{\mathrm{T}}J_{\mathrm{prior}}x \\
\phi_2(x,y) &= -\frac{1}{2}\|y - Cx\|^2 \\
\Phi_1(\theta_1) &= -\frac{1}{2}\log\det(\varphi J_{\mathrm{prior}}) \\
\Phi_2(\theta_2) &= -\frac{1}{2}\log\det(\gamma C^{\mathrm{T}}C)
\end{aligned}
\tag{46}
$$

where $\|\cdot\|^2$ denotes the standard squared Euclidean norm.

The E-step of the EM algorithm computes the expected values of the $\phi_i$ using the conditional mean of $x$ and the error covariances computed using the previous iteration's values of the parameter estimates. Some simple matrix manipulations yields the following:

$$
\begin{aligned}
\eta_1 \triangleq E[x^{\mathrm{T}}J_{\mathrm{prior}}x \mid y, \theta^{(n-1)}] \\
= \mathrm{tr}(J_{\mathrm{prior}}\widehat{P}^{(n-1)}) \\
+ (\hat{x}^{(n-1)})^{\mathrm{T}}J_{\mathrm{prior}}\hat{x}^{(n-1)}.
\end{aligned}
\tag{47}
$$

Due to the sparsity of $J_{\mathrm{prior}}$, we only need variances of individual nodes and covariances between pairs of neighboring nodes to perform this computation. Similarly

$$
\eta_2 \triangleq E[\|y - Cx\|^2\|y, \theta^{(n-1)}] = \|y - C\hat{x}^{(n-1)}\|^2 + \mathrm{tr}(C\widehat{P}^{(n-1)}C^{\mathrm{T}}), \tag{48}
$$

which can be computed from individual node variances. Thus all these quantities can be computed exactly or with accurate approximations using the inference algorithms described previously.

The M-step then computes updated estimates of the parameters. Thanks to the form of the log-partition function for our model, this maximization can be performed analytically, leading to the following expressions for the next parameter estimates:

$$
\varphi^{(n)} = \frac{N}{\eta_1}, \tag{49}
$$

$$
\gamma^{(n)} = \frac{N_{\mathrm{meas}}}{\eta_2}, \tag{50}
$$

where $N$ is the number of nodes in the graph, and $N_{\mathrm{meas}}$ is the number of measurements.

As the estimation of the noise variance is straightforward and standard, we illustrate only the result of estimating the parameter $\varphi$ controlling the level of smoothness in our prior. In this case, we really cannot think of obtaining multiple, independent samples of the underlying random field, although we can consider using multiple sets of *measurements* of that field. It is not difficult in this case to see that the estimation of $\varphi$ depends only weakly on the number of measurement sets available but much more strongly on the overall *size* of the underlying field, with

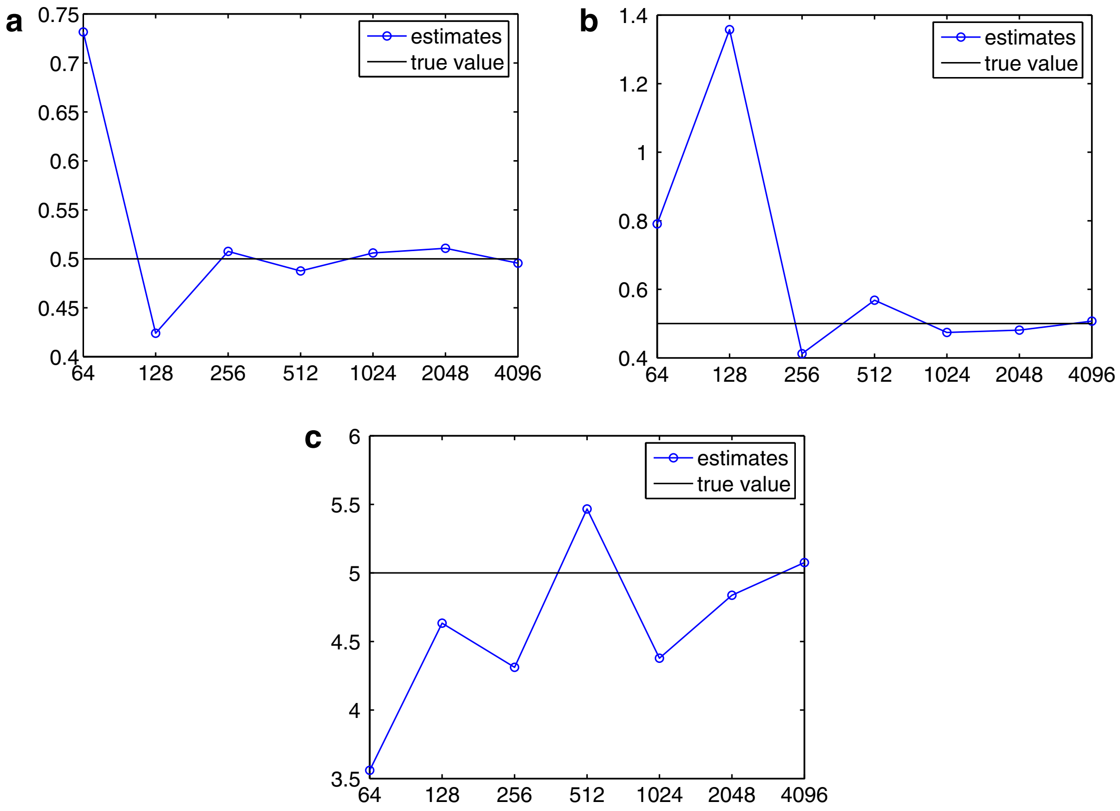


Fig. 12. Parameter $\varphi$ estimated from 5 sets of measurements generated by the finest scale nodes of the pyramidal graph. The $x$-axis show the number of nodes at the finest scale of the pyramidal graph: (a) true parameter: $\varphi = 0.5$, $\gamma = 2$; (b) true parameter: $\varphi = 0.5$, $\gamma = 0.1$; and (c) true parameter: $\varphi = 5$, $\gamma = 5$.

more accurate estimates resulting from data over larger fields. Fig. 12 illustrates this for three different examples of our model (corresponding to different choices for $\varphi$ and $\gamma$) where each curve displays estimates of $\varphi$ for fields of differing sizes (where for the larger-sized fields we used our adaptive multipole algorithm to compute estimates in each E step and the spliced wavelet method for computing the needed error variances and covariances).

## 4. Discussion

In this paper, we have introduced a class of Gaussian multiscale models defined on pyramidal lattices and described key properties of these models. Models of this type have been proposed before, but, thanks to very recent advances in inference for graphical models, we not only can provide additional motivation for these models but also develop a number of new and scalable algorithms for the solution of inference problems for these models.

One approach to the computation of optimal estimates for these models has conceptual ties to so-called multipole algorithms for the solution of partial differential equations. This algorithm takes advantage of the fact that long-distance correlations are mostly the result of the pyramidal structure of the models, while conditional correlations within scale are very local. These features led to an iterative algorithm that alternates global propagation of information using an embedded spanning tree and scale-by-scale local updates. Using recent results on so-called walk-sum analysis, such an algorithm is guaranteed to converge to the optimal estimates; moreover walk-sum analysis provides the basis for the adaptive choice of spanning trees and local updates at each iteration in order to achieve very rapid convergence. Furthermore, walk-sum analysis also leads to very efficient methods for so-called re-estimation problems in which we wish to modify an estimated field when local changes are made to the prior model or to the available data. A second approach to optimal estimation is based on so-called Lagrangian relaxation methods in which estimation is simultaneously carried out on a set embedded subgraphs (on each of which estimation is tractable), with coordination at each iteration to drive the separate inference results to common (and optimal) estimates and weighted error variances (that are upper bounds on the optimal values).

We have also presented two alternate methods for computing accurate approximations to the error variances in a fully scalable manner. Both of these make use of very recent results on so-called low-rank approximations using "spliced bases" to exploit the correlation structure of the field of interest in order to produce unbiased estimates of guaranteed accuracy. One of our approaches yields an iterative scale-by-scale method for approximation of variances that also exploits the fast correlation decay at any scale when conditioned on neighboring scales. The other exploits the long-distance correlation of the finest scale of our pyramidal models and uses spliced wavelet bases for very accurate approximations to the variances. We also demonstrated that the structure of our models and the scalable algorithms we have developed lead to a very efficient expectation-maximization algorithm for the estimation of model parameters.

This work suggests a variety of further lines of inquiry. Two such lines are related to the construction and meaning of the coarser scales in these pyramidal models. One of these focuses on giving physical meaning to the variables represented at these coarser scales. Indeed, some of the motivation we provided for our models comes from problems in which the available data on which we wish to base our model are themselves at multiple scales. Developing this idea will also allow our framework to seamlessly fuse data at multiple resolutions. An alternative point of view concerning the coarser scale variables is that they are there simply for statistical convenience – indeed we have shown that it is the inter-scale structure that captures long-distance correlations. As a result, it is of interest to consider designing these coarser scale variables and the relationships between scales in order to *simplify* the *intrascale* models – e.g., to *thin* the graphs within each scale in a principled manner. Such thinning methods are at the core of entropy-based methods such as those used in [2,15].

We anticipate that the investigations just described will have areas of overlap with wavelet-based representations – e.g., in which coarser variables represent sets of wavelet and scaling coefficients. Such an approach has been developed for purely quadtree models [35], and it is of interest to consider its extension to this richer framework. In addition, using wavelets may open up the possibility of developing new algorithms for non-linear *wavelet cascade* models [39] that are capable of directly capturing abrupt changes – e.g., due to edges in images or cliffs in surfaces. More generally, the extension of these ideas to pyramidal graphical models with discrete states or non-Gaussian variables is of considerable interest and applicability, something that can already be seen in the literature [3–8,33]. Developing methods for constructing such models based on maximum entropy principles and then developing inference algorithms exploiting the advances described in this paper represent exciting and promising avenues for further work.

## References

[1] A.S. Willsky, Multiresolution Markov models for signal and image processing, Proc. IEEE 90 (August) (2002) 1396–1458.

[2] J.K. Johnson, A.S. Willsky, A recursive model-reduction method for estimation in Gaussian Markov random fields, IEEE Trans. Image Process 17 (1) (2008) 70–83.

[3] C. Graffigne, F. Heitz, P. Perez, F. Prêteux, M. Sigelle, J. Zerubia, Hierarchical Markov random field models applied to image analysis: a review, in: SPIE Conference on Neural, Morphological Stochastic Methods in Image Signal Processing, vol. 2568, 1995, pp. 12–17.

[4] F. Heitz, P. Perez, P. Bouthemy, Multiscale minimization of global energy functions in some visual recovery problems, in: CVGIP: Image Understanding, vol. 59, no. 1, 1994, pp. 125–134.

[5] Z. Kato, M. Berthod, J. Zerubia, Multiscale Markov random field models for parallel image classification, in: Proceedings ICCV, Berlin, May 1993.

[6] Z. Kato, M. Berthod, J. Zerubia, A hierarchical Markov random field model and multitemperature annealing for parallel image classification, Graph. Model Image Process. 58 (1) (1996) 18–37.

[7] Z. Kato, J. Zerubia, M. Berthod, Unsupervised parallel image classification using Markovian models, Pattern Recogn. 32 (4) (1999) 591–604.

[8] J. Li, R.M. Gray, R.A. Olshen, Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models, IEEE Trans. Image Process. 46 (August) (2000) 1826–1841.

[9] B. Bollobás, Modern Graph Theory, Springer, 1998.

[10] R. Diestel, Graph Theory, Springer, 2000.

[11] M.I. Jordan, An Introduction to Graphical Models, MIT Press, in press.

[12] S.L. Lauritzen, Graphical Models, Oxford University Press, Oxford, UK, 1996.

[13] M.I. Jordan, Graphical models, Stat. Sci. 19 (2004) 140–155 (Special Issue on Bayesian Statistics).

[14] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, San Mateo, CA, 1988.

[15] J.K. Johnson, V. Chandrasekaran, A.S. Willsky, Learning Markov structure by maximum entropy relaxation, in: 11th International Conference on Artificial Intelligence and Statistics (AISTATS'07), San Juan, Puerto Rico, March 2007.

[16] T. Kailath, A.H. Sayed, B. Hassibi, Linear Estimation, Prentice Hall, 2000.

[17] K. Plarre, P. Kumar, Extended message passing algorithm for inference in loopy Gaussian graphical models, Ad Hoc Networks (2004).

[18] D.M. Malioutov, J.K. Johnson, A.S. Willsky, Walk-sums and belief propagation in Gaussian graphical models, J. Mach. Learning Res. 7 (October) (2006) 2003–2030.

[19] Y. Weiss, W.T. Freeman, Correctness of belief propagation in Gaussian graphical models of arbitrary topology, Neural Computat. 13 (2001) 2173–2200.

[20] M.R. Luettgen, W.C. Karl, A.S. Willsky, R.R. Tenney, Multiscale representations of Markov random fields, IEEE Trans. Signal Process. 41 (December) (1993) 3377–3396.

[21] J.K. Johnson, Estimation of GMRFs by recursive cavity modeling, Master's thesis, MIT, Cambridge, MA, March 2003.

[22] J.K. Johnson, D.M. Malioutov, A.S. Willsky, Walk-sum interpretation and analysis of Gaussian belief propagation, in: advances in Neural Information Processing Systems, vol. 18, 2006, pp. 579–586.

[23] V. Chandrasekaran, J.K. Johnson, A.S. Willsky, Estimation in Gaussian graphical models using tractable subgraphs: A walk-sum analysis, IEEE Trans. Signal Process., in press.

[24] E.B. Sudderth, M.J. Wainwright, A.S. Willsky, Embedded trees: estimation of Gaussian processes on graphs with cycles, IEEE Trans. Signal Process. 52 (11) (2004) 3136–3150.

[25] J.K. Johnson, D.M. Malioutov, A.S. Willsky, Lagrangian relaxation for MAP estimation in graphical models, in: 45th Annual Allerton Conference on Communication, Control and Computing, 2007.

[26] M.J. Wainwright, T.S. Jaakkola, A.S. Willsky, A new class of upper bounds on the log partition function, IEEE Trans. Inform. Theory 51 (7) (2005) 2313–2335.

[27] M.J. Wainwright, T.S. Jaakkola, A.S. Willsky, MAP estimation via agreement on (hyper) trees: Message-passing and linear-programming approaches, IEEE Trans. Inform. Theory 51 (11) (2005) 3697–3717.

[28] V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization, IEEE Trans. on Pattern. and Mach. Intell. 28 (10) (2006) 1568–1583.

[29] J.K. Johnson, Convex relaxation methods for graphical models, MIT, Cambridge, MA, in preparation.

[30] M.J. Choi, Multiscale Gaussian graphical models and algorithms for large-scale inference, Master's thesis, MIT, Cambridge, MA, May 2007.

[31] D.M. Malioutov, J.K. Johnson, A.S. Willsky, Low-rank variance estimation in large-scale GMRF models, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, May 2006.

[32] D.M. Malioutov, J.K. Johnson, A.S. Willsky, GMRF variance approximation using spliced wavelet bases, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, Honolulu, Hawaii, April 2007.

[33] B. Gidas, A renormalization group approach to image processing problems, IEEE Trans. Pattern Anal. Mach. Intell. 11 (2) (1989) 164–180.

[34] W.L. Briggs, A Multigrid Tutorial, SIAM, Philadelphia, PA, 1987.

[35] K. Daoudi, A.B. Frakt, A.S. Willsky, Multiscale autoregressive models and wavelets, IEEE Trans. Inform. Theory 45 (3) (1999) 828–845.

[36] P.W. Fieguth, W.C. Karl, A.S. Willsky, Efficient multiresolution counterparts to variational methods for surface reconstruction, Comput. Vision Image Underst. 70 (2) (1998) 157–176.

[37] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348.

[38] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum-likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc. 39 (1977) 1–38.

[39] M.J. Wainwright, E.P. Simoncelli, A.S. Willsky, Random cascades on wavelet trees and their use in modeling natural images, Appl. Computat. Harmonic Anal. 11 (2001) 89–123.