Multiscale Stochastic Realization and Model Identification with Applications to Large-Scale Estimation Problems

by

William W. Irving

S.B., Massachusetts Institute of Technology (1987)
S.M., Massachusetts Institute of Technology (1991)
E.E., Massachusetts Institute of Technology (1992)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1995

Author .....
Department of Electrical Engineering and Computer Science
August 25, 1995

Certified by .....
Alan S. Willsky
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Frederick R. Morgenthaler
Graduate Officer, Department of EECS

Multiscale Stochastic Realization and Model Identification with Applications to Large-Scale Estimation Problems

by

William W. Irving

Submitted to the Department of Electrical Engineering and Computer Science
on August 25, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering

Abstract

A substantial challenge in signal processing is to devise estimation algorithms for 2-D random fields that are both computationally efficient and statistically optimal. Typical approaches, such as those related to Markov random fields, are iterative, slow and do not yield error covariance information. In contrast, we apply, refine and extend a recently developed statistical framework that overcomes these difficulties. Central to this framework is a class of multiscale stochastic processes, indexed by the nodes of a pyramidal tree structure, that evolve according to scale-recursive dynamics which are much like the time-recursive dynamics of Gauss-Markov time-series models.

We develop a theory for multiscale stochastic realization that represents a generalization of Akaike's canonical correlations approach to stochastic realization of time series. Our extension of the time-series ideas is non-trivial, because the multiscale process state must act as an interface among three or more subsets of the process, not just two. We demonstrate the utility of our realization theory by building multiscale models for random fields, and subsequently applying these models to solve some challenging 2-D estimation problems that are impractical to address with FFT-based estimation methods.

We treat an important problem in automatic target recognition with synthetic aperture radar (SAR). From actual SAR imagery, we identify a pair of multiscale models that capture the characteristically distinct scale-to-scale variations in speckle pattern for imagery of man-made objects and of natural clutter, respectively. We incorporate these models into a new algorithm for discriminating between imagery of man-made objects and of natural clutter. Application of this algorithm to an extensive dataset of actual SAR imagery leads to substantial and statistically significant improvement in receiver operating characteristics, compared to a standard, established discriminator developed at MIT Lincoln Laboratory.

Finally, we extend the multiscale framework to allow for models in which distinct nodes on a given level may correspond to overlapping portions of the image domain. We then build so-called overlapping-tree models, using our established realization techniques. These models lead to an elegant way to overcome the visually distracting blocky artifacts that are typical of estimates produced by standard multiscale models. Although the simpler post-processing technique of low-pass filtering can also eliminate this blockiness, such filtering can destroy error covariance information provided by the estimation algorithm, and can limit the resolution of fine-scale details. In contrast, our overlapping-tree approach allows for efficient calculation of both error covariance information and nearly optimal, smooth estimates, with fine-scale detail preserved.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering

Acknowledgments

Attending MIT graduate school is a phenomenal experience. All throughout my secondary and undergraduate school days, I thought that *getting an education* meant slogging one's way through some coursework and homework, in order to get good grades, so that one could slog his way through more of the same. I was wrong. I have since learned that learning can be a much more personal, human endeavor; initiative and hard work are important, but more so are the interactions and collaborations with others. With regard to these *others*, my tremendous good fortune has been to be surrounded by top-notch advisors, teachers, colleagues and friends. I hope that I can at least partially repay these people for what they have given me, wittingly or otherwise, by acknowledging them here.

First, I wish to acknowledge and thank my doctoral thesis advisor, Professor Alan Willsky. Alan's energy and enthusiasm go virtually unrivaled, and anyone who meets him cannot help but detect his lightning fast intellect. In spite of his stature, he still has not forgotten the importance of his students, and his boundless support, both for my research ideas and for me as an individual, have been essential to my finding the stamina to complete the doctoral program. I am not sure that Alan has quite left the imprint he would like on my writing, as he tries to rein in my (sometimes overly dramatic) flair, but I thank him for trying.

Second, I wish to thank Professor Clem Karl, who served on my thesis committee, but more importantly, who served as an extremely accessible mentor. No matter how busy Clem was, he never turned me away from his door; I could always count on him to provide advice and to share his perspective on any technical problem, or on any aspect of life at MIT or life in general. I thank also the other members of my thesis committee, John Tsitsiklis and Munther Dahleh, for their helpful comments.

After obtaining my bachelor's degree from MIT in 1987, I worked full time for two years at MIT Lincoln Laboratory. These years played a fundamental role in shaping my decision to return to graduate school. My Lincoln colleagues Rick Barnes and Dennis Blejer were particularly influential. Rick taught me the essentials of detection and estimation theory, and was an enviable master at back-of-the envelope calculations. Dennis taught me the essentials of electromagnetic wave theory and also the beauty of cycling through the hills of Vermont.

Lincoln fostered my move back to graduate school, by generously funding my studies through their Staff Associate Program. For that support, I am indebted to my Lincoln group leader, Jerry Morse. Actually, Jerry not only had enough faith in my abilities to send me back to school, but he also gave me virtually complete freedom to pursue my own research interests during my yearly summer retreats back to Lincoln. These retreats were unusually productive, thanks in large part to the influence of my Lincoln colleague Les Novak. Les's inimitable style prodded me to carry projects from start to finish, during each of the five graduate-school summers I spent at Lincoln. These projects were always challenging, and provided a refreshing change from my work on campus.

Over the years, I have come to place a great deal of importance on clear writing and crisp, effective communication in general. Although I cannot vouch for my ability to execute either, I *can* credit my very good friend and Lincoln colleague Shawn Verbout for shaping my perspective on these matters. Shawn is a superb technical writer, and our summer collaborations have given me a chance to see his secret. The fact is, there is no secret; his good writing comes from painstaking, hard work, which I now try to bestow on my own writing and speaking endeavors. Outside of the workplace, Shawn has been simply a good

friend, with whom I have shared many interesting conversations at good restaurants all over Boston, as well as trips to Atlantic City and trips to the driving ranges and golf courses.

Among my fellow students at MIT, I wish to thank Paul Fieguth for the technical collaboration and for the extensive LaTeX advice. I also thank Mike Daniel for the very careful reading of Chapter 3, for his intolerance of my sometimes vague mathematical arguments, and for the good times on the ski slopes of Colorado. My office mates, past and present, certainly deserve mention: Mike Branicky, Charlie Fosgate, Rachel Learned, Cedric Logan, Mark Luettgen, Eric Miller, Venkatesh Saligrama, Mike Schneider, and Ted Theodosopoulos. Thanks also go to the fellow travelers to France two years ago: Mickey Bhatia and Seema Jaggi.

For several years, I rented floors of one house or another, sharing the expenses and space with some terrific, interesting people. Particularly noteworthy is Anil Duggal, with whom I lived with for five years, and who remains a steadfast friend. Also, I wish to mention Al Cangahuala, who has been a very close friend ever since our Chi Phi (and air band) days of the mid-eighties.

Having a supportive family is key to surviving a long haul like graduate school, and my family has been with me the whole way: Laura, Mom, and Dad, I thank you all.

Finally, I must express my gratitude to Bruno Suard, a fellow LIDS student, who on December 5, 1991, suggested that we go to the Roxy, since he knew "some German chick" who was saving him some tickets for free admission. Well, we went; I met the German woman, Anneli Mynttinen, and now she is my very dear wife. Anneli deserves a lot of credit for supporting me emotionally, especially during the final haul of the past three months.

Contents

List of Figures

List of Tables

Notational Conventions

Unless stated explicitly otherwise, all vectors are assumed to be column vectors. We refer to a matrix consisting entirely of zeros by $\mathbf{0}$. For any set $\mathcal{A}$, we denote the cardinality of $\mathcal{A}$ by $|\mathcal{A}|$.

Letting x and y be random vectors (which, in accordance to our just established conventions, are *column* vectors), we denote the expect value of x by $E(x)$, and the covariance of x by P_x,

$$P_x \;=\; E\left[(x - E(x))\,(x - E(x))^T\right].$$

In a slight departure from traditional usage, we always use the notation $E(x|y)$ to mean the linear least-squares estimate of x given y; in the special case that x and y are jointly Gaussian, this convention coincides with the more traditional meaning of $E(x|y)$ as the expected value of x conditioned on y. We use the notation $x \perp y$ to mean that x and y are uncorrelated,

$$x \perp y \Longleftrightarrow E\left[(x - E(x))\,(y - E(y))^T\right] \;=\; \mathbf{0}.$$

Finally, we will find utility in having special notation to describe algorithm complexity. To develop this notation, we here suppose that $f(\cdot)$ and $g(\cdot)$ are functions from the positive integers to the positive reals. Then, we write $f(n) = \mathcal{O}(g(n))$ if there exists a constant $c > 0$ such that, for large enough n, $f(n) \leq cg(n)$. This convention makes precise, for instance, the colloquial statement that for a matrix of dimensions $n \times n$, *matrix inversion requires* $\mathcal{O}(n^3)$ *floating point operations.*

Other, more specialized notational conventions will be introduced as they are needed in the body of the text.

Chapter 1

Introduction

A longstanding and substantial challenge in signal processing has been to devise algorithms for statistical inference that are not only optimal mathematically, but also efficient computationally. There are a variety of important tasks that call for such algorithms: (i) Bayesian least-squares estimation, (ii) error covariance calculation, (iii) spectral estimation, (iv) likelihood calculation, (v) system identification and (vi) Monte-Carlo simulation via sample path generation. Indeed, the development of algorithms for all these tasks is fundamental to both statistical processing of 1-D signals and also of 2-D signals and images. However, in spite of this common importance to both 1-D and 2-D, there is a rather sharp disparity in the availability of optimal, efficient algorithms for the two cases.

While in the 1-D context, there exists an extensive, well-understood body of literature, documenting optimal, efficient algorithms for carrying out all of the aforementioned tasks, in the 2-D context, the literature is not as coherent. *Almost without exception, 2-D statistical inference problems that are encountered in practice are substantially more challenging than their 1-D counterparts.* Even the seemingly most innocuous 2-D problem, such as estimating a signal in noise, can require a tremendous computational burden for optimal solution, even though the 1-D counterpart may be manageable, or even trivial, to solve optimally. This greater challenge of 2-D is not due merely to the fact that 2-D problems involve more variables or unknowns. The deeper reason, which actually poses a more surmountable, concrete challenge, is that the signal processing community has had more success in devising good mathematical models for 1-D signals than for 2-D ones. In particular, Gauss-Markov time-series models and the associated Kalman filter have played central roles in the success of 1-D signal processing. On the other hand, attempts to extend this success to 2-D have traditionally been unsatisfactory. For instance, Markov random fields (MRFs) are suitable for modeling a rich class of 2-D random phenomena, but do not generally lead to fast estimation algorithms. This general absence of good 2-D statistical models has lead to shortcomings in the design of good 2-D algorithms; frequently, there is a perceived need to compromise statistical consistency and optimality in order to keep computational costs to a reasonable level.

To be sure, the demand for good 2-D algorithms is not purely aesthetic, nor is it merely an antidote for theorists' desire for ever more general results. Rather, the demand is practical, and in fact one does not have to search far to find 2-D problems, having a current, real context, that could benefit substantially from such algorithms. For instance, there are many remote sensing studies of the earth's terrain and environment in which the key challenge is to assimilate efficiently large amounts of measured data. The need for data assimilation tools

is actually quite acute with imagery produced, for example, by synthetic aperture radar (SAR) systems [16, 51], especially as high-resolution SARs become increasingly common. Operating in a so-called stripmap mode, airborn and spaceborn SAR sensors can generate (in a matter of minutes) imagery representing several square kilometers, at resolutions sometimes finer than one meter squared. For applications including image compression, terrain identification, image segmentation, and discrimination between targets and clutter, the great amount of data calls for fundamental, systematic image analysis techniques.

In light of the genuine need for good 2-D statistical models and algorithms, it is both satisfying and exciting to note that over the past five years, the gap between 1-D and 2-D has narrowed. This progress is due in part to a new, powerful framework for statistical signal processing that was first introduced in [13–15], and has been more recently applied, refined, and extended by the work in this thesis and also in [22, 23, 34, 35, 43–46]. This framework represents a genuinely successful extension of the Gauss-Markov/Kalman-filtering approach from 1-D to 2-D, and has demonstrated utility in confronting data assimilation problems of dauntingly large dimension; two such successes include calculation of optical flow [46] and smoothing of ocean altimetric data [22]. In the latter work, for example, the authors were able to estimate both ocean surface height *and* associated error statistics for a 512×512 grid, all in one minute on a current-generation single-processor workstation.[1]

1.1 A Representative Problem: Linear Least-squares Estimation

To supply a heightened perspective regarding the role played by this new statistical framework, let us consider a representative problem to which the framework is applicable. The particular problem we consider is linear least-squares estimation with error covariance calculation. Patterning our discussion after [22], we begin by examining the success of the Gauss-Markov estimation framework for 1-D problems. Next, we examine MRFs, both causal and non-causal, viewing them as an attempt to extend this success to 2-D. Finally, having established an appropriate context, we introduce the new framework in which we will be working throughout the rest of the thesis; we describe the structure of the class of stochastic models around which the framework is built, and we comment on some of the statistical inference algorithms these models admit.

1.1.1 Problem Formulation

Let us suppose we have a grid of points $p_1, p_2, \ldots, p_N$, indexing a discretization of either space or time. Associated with each point p_i is a random signal value $x(p_i)$, having dimension d, and a noise-corrupted measurement $y(p_i)$,

$$y(p_i) \;=\; C(p_i)x(p_i) + v(p_i), \quad i = 1, 2, \ldots, N.$$

In this relation, $C(p_i)$ is a matrix of appropriate dimension,[2] while $v(p_i)$ represents the measurement noise or error, assumed to be zero-mean, white and uncorrelated with the

[1] The specific workstation model used was a Sparc-10.

[2] We can accommodate the case of no measurement at point p_i by setting $C(p_i)$ equal to a row vector consisting of all zeros.

signal:

$$E\left(v(p_i)\right) = \mathbf{0}, \quad \forall i \tag{1.1}$$

$$E\left(v(p_i)v^T(p_j)\right) = \begin{cases} R(p_i) & \text{if } i = j \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{1.2}$$

$$E\left(x(p_i)v^T(p_j)\right) = \mathbf{0}, \quad \forall j, \tag{1.3}$$

with $R(p_i)$ assumed to be positive definite, for $i = 1, 2, \ldots, N$. For convenience, we consolidate the quantities of interest in this problem into appropriate block-partitioned vectors and matrices. In this more compact notation, x denotes the vector of signal values $x(p_i)$ ordered sequentially

$$x \equiv \left(\begin{array}{cccc} x^T(p_1) & x^T(p_2) & \cdots & x^T(p_N) \end{array} \right)^T,$$

while y and v are defined analogously in terms of the observation and noise terms $y(p_i)$ and $v(p_i)$, respectively. The covariance of x is denoted by P_x, which is also assumed to be positive definite. Finally, to be consistent with (1.1)-(1.3), the relation between x and y is given as

$$y = Cx + v, \quad E(v) = \mathbf{0}, \quad E(vv^T) = R \quad E(xv^T) = \hat{\mathbf{0}}, \tag{1.4}$$

with

$$\begin{aligned} C &\equiv \text{diag}\left(C(p_1), C(p_2), \ldots, C(p_N)\right), \\ R &\equiv \text{diag}\left(R(p_1), R(p_2), \ldots, R(p_N)\right). \end{aligned}$$

Our objective is to determine the linear least-squares estimate of x given y. Assuming that x is zero-mean,[3] this estimate has the form $\hat{x}(y) = Ly$ where L is chosen to minimize the mean-square error $E\left[(x - \hat{x}(y))^T (x - \hat{x}(y))\right]$. To allow for reasonable assessment of the accuracy of this estimate, we also seek the estimate's error covariance $P_{\tilde{x}}$,

$$P_{\tilde{x}} \equiv E\left[(x - \hat{x}(y))(x - \hat{x}(y))^T\right],$$

or at least the diagonal block-components of $P_{\tilde{x}}$.

1.1.2 Stationarity and the FFT

If x represents a stationary process or field having periodic boundary conditions (i.e., a 1-D process defined on a circle or a 2-D field defined on a toroid), and additionally, the measurements in y are dense and of uniform quality (implying that both C and R are multiples of the identity matrix), then P_x and $P_{\tilde{x}}$ both have the same eigenvectors, which happen to be closely related to the FFT. As detailed in Chapter 2, this close relationship implies that the computational efficiencies of the FFT can be brought to bear to calculate both $\hat{x}(y)$ and $P_{\tilde{x}}$.

As one might expect, however, there are many practical problems, such as the oceanographic application, containing exacerbating factors that destroy the symmetry and unifor-

[3] Otherwise, we can first subtract out its mean m_x, and simply add it back after estimation of $(x - m_x)$.

mity needed for FFT techniques: (i) a non-stationary sensed phenomenon, (ii) occasional data dropouts, (iii) non-uniform measurement quality, (iv) an ocean that is not toroidal. Thus, a stationarity assumption does not always make sense, and the FFT does not completely mitigate the computational challenges of this problem.

1.1.3 Implicit Statistical Description of x

If x represents a non-stationary process or field, then for problems of practical size, there is little sense or value in attempting to specify explicitly the full covariance matrix P_x, unless this matrix is extremely sparse. While such sparse structure will certainly be present in phenomena exhibiting only local correlations, where a banded covariance matrix is appropriate, there are many other phenomena, such as fractal phenomena, that exhibit important long-range correlations.

A standard, compact way to describe both stationary *and* non-stationary phenomena, possibly having important correlations at many scales, is to use an *implicit* model for the statistical structure of x. The implicit models we consider have the general form

$$Gx = w, \tag{1.5}$$

where G is a (generally sparse) matrix or *model*, x is the signal and w is the driving noise. If G is invertible, then by letting P_w be the covariance of w, (1.5) can be recast as

$$P_x^{-1} = G^T P_w^{-1} G. \tag{1.6}$$

As we will see, if certain additional structure is imposed on G and P_w^{-1}, then we obtain a special class of implicit models that has long underpinned the success of 1-D estimation methods. While attempts to extend this success to 2-D have traditionally been unsatisfactory, we will ultimately see that implicit models of the form (1.5) and (1.6) can also be specialized to yield the class of models fundamental to the new estimation framework introduced in [13–15] and explored in this thesis. With these models, we can indeed manage large, non-stationary 2-D estimation problems, involving possibly sparse measurements of non-uniform quality.

1.1.4 The Normal Equations

To see the kind of structure needed in G and P_w^{-1} (see (1.5) or (1.6)) to simplify the calculation of the least-squares solution, let us examine closed-form, analytical expressions for both $\hat{x}(y)$ and $P_{\tilde{x}}$. Actually, there are many equivalent ways to express these two quantities; the most useful and insightful one for our purposes is given implicitly by the following so-called *normal equations*:

$$P_{\tilde{x}}^{-1}\hat{x}(y) = C^T R^{-1} y, \tag{1.7}$$

$$P_{\tilde{x}}^{-1} = P_x^{-1} + C^T R^{-1} C = G^T P_w^{-1} G + C^T R^{-1} C \tag{1.8}$$

If these equations are solved without any attempt to discern and exploit special structure they may have, then a general-purpose algorithm such as Gaussian elimination will be required, leading to a computational cost of $\mathcal{O}((Nd)^3)$ arithmetic operations. For large-scale data assimilation problems, such as the oceanographic application cited earlier where $N \approx 2.5 \times 10^5$, this general approach becomes absolutely prohibitive.

A closer inspection of (1.7) and (1.8) reveals that the critical factor determining the possibility of more specialized, efficient approach is the structure of the *inverse* of the prior covariance or equivalently the structure of $G^T P_w^{-1} G$. Specifically, since C and R are block-diagonal, it immediately follows that $C^T R^{-1} C$ is block-diagonal, and hence, thanks to (1.8), any off-diagonal sparse and regular structure that is present in P_x^{-1} will be preserved in $P_{\hat{x}}^{-1}$; in turn, this structure can be exploited in the solution of (1.7), to yield $\hat{x}(y)$.

1.1.5 Gauss-Markov Time-series Models

Let us now consider how we can structure G and P_w^{-1} so that we simultaneously achieve two effects: (i) the resulting model class is rich statistically, and (ii) the solution to (1.7), using (1.8), is computationally inexpensive. If both G and P_w are constrained to be block diagonal, then we certainly achieve (ii), but we fail to achieve (i). However, by just slightly relaxing this rigid, block-diagonal structure, we are led to Gauss-Markov time-series models, which constitute the most popular and powerful class of implicit models for 1-D applications. These models achieve both of our desired effects by allowing G to be *lower bidiagonal*, while continuing to constrain P_w to be block diagonal. In particular, the dynamics of these processes are governed by a specialization of (1.5) in which the sequence of components of x, which we now denote by $x(1), x(2), \ldots$, obey a white-noise driven difference equation, evolving in discrete time:

$$x(n+1) = A(n)x(n) + w(n) \tag{1.9}$$

$$y(n) = C(n)x(n) + v(n). \tag{1.10}$$

Here, $A(i)$ represents a one-step transition matrix, and $w(i)$ is the noise driving term, assumed to be zero-mean, white and uncorrelated with both the signal and the observation noise. By defining the vector w as

$$w \equiv \begin{pmatrix} x^T(1) & w^T(1) & w^T(2) & \cdots & w^T(N-1) \end{pmatrix}^T,$$

it becomes clear that (1.9) is a special case of (1.5) in which G has the following form:

$$G = \begin{pmatrix} I & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -A(1) & I & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -A(2) & I & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -A(3) & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -A(N-1) & I \end{pmatrix}.$$

It turns out that for any WS stationary process having a rational spectrum, there is a corresponding time-series model of the form (1.9), and for many non-stationary processes, there is also a model of the form (1.9). Thus, Gauss-Markov time-series models represent a specialization of (1.5) that is still rich enough to capture the statistical behavior of a wide variety of phenomena. In fact, these models have been successfully used for statistical modeling in many diverse fields, including economics, guidance control, and speech processing. Equally importantly, these models admit efficient algorithms for many signal processing tasks, including the calculation of both $\hat{x}(y)$ and the diagonal blocks of $P_{\hat{x}}$. Indeed, for any given Gauss-Markov model, the justly celebrated Kalman filter [5] and the associated

Rauch-Tung-Striebel smoother [58] can be used to recursively and efficiently obtain $\hat{x}(y)$ and the diagonal block elements of $P_{\tilde{x}}$. The plausibility of this fact can be seen by noting from (1.6) that P_x^{-1} is tridiagonal, and therefore, in light of (1.8), this tridiagonal structure is preserved in $P_{\tilde{x}}$. Systems of linear equations involving tridiagonal matrices are solvable in $\mathcal{O}(Nd^3)$ operations, or equivalently $\mathcal{O}(d^3)$ per-pixel operations, which is exactly the complexity of the Kalman filter. Since the per-pixel complexity does not grow with N, this complexity is quite attractive computationally.

There are other attractive features of both Gauss-Markov models and the concomitant Kalman filter. For one, the recursion in (1.9) is easily implemented on a computer, thus providing an efficient method for sample-path generation of time series. Also, though not obvious from our analysis and comments, the Kalman filter can be used to *whiten* efficiently a sequence of measurements $y(1), y(2), \ldots$, and the resulting whitened sequence can in turn be used for efficient likelihood calculation, assuming that the random variables are Gaussian. Since likelihood calculations are instrumental in parameter identification and system identification, the Kalman filter leads to efficient techniques for confronting these other statistical inference problems as well.

1.1.6 Markov random fields

Given the power and success of the 1-D Gauss-Markov modeling and estimation framework, it is natural to hope this success can be extended to 2-D. Ideally, such an extension should yield a model class that is rich statistically and for which efficient estimation algorithms can be devised. These dual requirements pose a tall order, and in fact most approaches documented in the literature fail to satisfy one or the other.

One immediate difficulty is that most 2-D processes and images lead to a sequence $x(p_1)$, $x(p_2), \ldots, x(p_N)$ having no natural causal structure, regardless of the spatial arrangement of the points $p_1, p_2, \ldots, p_N$ in the Cartesian plane; consequently, there is a seeming mismatch between the dynamical relation in (1.9) and 2-D processes. Nevertheless, there is no intrinsic mathematical difficulty with developing classes of 2-D random fields having causal structure, as exemplified by the theories of Markov mesh random fields [1, 18], and their generalization, nonsymmetric half-plane Markov chains [18, 37, 66]. In the latter, for instance, a lexicographic ordering is imposed on the random field; if this field has dimensions, say, $N \times$N, then a state vector of the form

$$\begin{aligned}\mathbf{x}(m,n) \;=\; \Big(\; & x(m,n) \quad x(m-1,n) \quad \cdots \quad x(1,n) \quad x(N,n-1) \quad \cdots \\ & \cdots \quad x(1,n-1) \quad \cdots \quad x(N,n-K) \quad \cdots \quad x(m-K,n-K) \;\Big)^T,\end{aligned}$$

can be defined, for some K, such that $\mathbf{x}(m,n)$ is related to its lexicographic predecessor by a difference equation of the form (1.9). Since the required state dimension is proportional to the width of the image, unduly high dimensions are required. Combining this difficulty with undesirable anisotropies that result from the causal ordering, we conclude that the causal MRF formalism is not very satisfactory for solving practical least-squares problems.

The problem of causality can be eliminated by considering another class of MRFs that still have the form (1.5), but now with a neighborhood structure that is spatially symmetric. Unlike their causal counterparts, these MRFs are well-suited to model a rich class of natural phenomena, and can capture the spatial continuity that is characteristic of many images. A detailed discussion of the some of the properties of these MRFs is contained in Chapter 2.

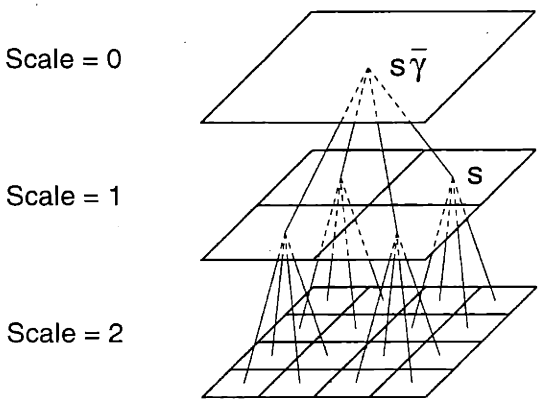


Figure 1-1: Illustration of the first three levels of a quadtree, which is useful for indexing multiscale representations of random fields. The state at each node represents an aggregate description of a subset of the finest-scale random field; this correspondence between nodes and field subsets is denoted in the figure by the quadrantal boundaries. The parent of node s is denoted by $s\bar{\gamma}$.

For our purposes here, there are two important facts that emerge from that discussion, which together imply that these MRFs do *not* generally admit efficient procedures for calculating $\hat{x}(y)$ and $P_{\tilde{x}}$. For one, when the matrix G is placed in (1.5), the resulting system of linear equations is structurally identical to the equations resulting from discretization of an elliptic partial differential equation (PDE), with G being symmetric, positive definite and extremely sparse; for this reason, the matrix G is sometimes referred to as an *elliptic operator*. Second is the somewhat surprising fact that $P_x^{-1} = G$. Now, in light of (1.8), the matrix $P_{\tilde{x}}^{-1}$ is also an elliptic operator, and hence, solving (1.7) for $\hat{x}(y)$ is computationally equivalent to solving an elliptic PDE, while inverting $P_{\tilde{x}}^{-1}$ is equivalent to an inverting an elliptic operator. While these elliptic PDEs can often be solved by efficient iterative procedures, such as successive-overrelaxation or multigrid approaches, the a significant difficulty is that error-covariance calculation (i.e., the calculation of the diagonal blocks of $P_{\tilde{x}}$) must be carried out in addition to solving the elliptic PDE. This additional computational cost is at least as great as, and typically much greater than, the cost of solving the PDE. Adding to these computational difficulties is that fact that MRFs do not admit efficient algorithms for either likelihood calculation or sample-path generation. We conclude from these comments that MRFs do not provide a completely satisfactory framework for statistical inference.

1.1.7 A New Approach via Multiscale Modeling

The 2-D extensions described so far have suffered either from being too artificial (i.e., causal MRFs) or from admission of only computationally complex algorithms (i.e., non-causal MRFs). Now, finally, we describe an extension that succeeds in overcoming these deficiencies, thereby bringing to 2-D the full set of advantages of the Gauss-Markov 1-D framework. The key to this newfound success is its broadening of scope to consider stochastic processes not just at a single resolution scale, but at a whole sequence of resolution scales. More specifically, this new estimation framework is built around a class of stochastic processes that evolve in scale, where the scale-recursive dynamics underlying the process evolution represent a clear intellectual descendant of the more traditional time-recursive dynamics underlying the evolution of Gauss-Markov processes. Effectively, this framework expresses and exploits the time-like nature of scale.

Figure 1-1 illustrates a quadtree structure, which is often used to index multiscale repre-

sentations of random fields. Each level of this tree corresponds to a particular scale m, with larger m corresponding to finer resolution. The state $x(s)$ at any given node s represents an appropriate, aggregate description of the subset of the finest-scale process that descends from the given node. Letting $s\bar{\gamma}$ denote the parent of a given node s, the dynamics underlying the process evolution are described by the following first-order vector-valued difference equation:

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

In analogy with (1.9), $A(s)$ and $B(s)$ are matrices of appropriate size; the recursion is initialized at the root node, $s = 0$, with a state variable $x(0)$, and the term $w(s)$ represents white driving noise, which is both zero mean and uncorrelated with the initial condition $x(0)$. A much more detailed discussion of these processes is contained in Chapter 2.

Why multiresolution?

There are at least three distinct ways in which multiresolution concepts may enter a 2-D estimation problem, thereby providing a natural fit between the given problem and our estimation framework. First, the *phenomenon* under investigation may display important features at multiple resolutions. For example, in the context of remote sensing, the work in [47,55] suggests that natural terrain imaged by a sensor may be fruitfully interpreted as a superposition of fine resolution features on a more coarsely varying background.

Second, whether or not the underlying phenomenon is deemed to have important multiresolution features, the *data* may be of varying resolution. For example, a well-known difficulty of SAR sensing is that high resolution requires high frequency while foliage penetration requires lower frequency. Thus there is tradeoff between resolution and foliage penetration capabilities, which is perhaps best confronted by using a suite SAR sensors, each operating at a distinct frequency. If this strategy is employed, then a need arises for systematic techniques for fusing high-resolution, high frequency data with lower resolution, low-frequency data.

Third, whether or not the phenomena or data are deemed to have important multiresolution features, there are still good reasons for the *algorithm* to be multiresolution. There is something particularly evocative about proceeding by the divide-and-conquer route that a multiresolution algorithm affords, even if no other aspect of the given problem has salient multiresolution features. In fact, much of the work in this thesis, is directed toward problems that fall into this last category, wherein all attention focuses on the finest-scale, while the multiresolution structure of both the model and processing is viewed merely as a convenient computational vehicle.

Advantages of multiresolution framework

One of the primary reasons the framework is useful is that it leads to extremely efficient, statistically optimal algorithms for signal and image processing. These algorithms exploit the special statistical structure of our models in much the same way that the Kalman filter exploits the structure of Gauss-Markov time-series models. In fact, a particularly successful example of a multiscale-based estimation algorithm is a direct generalization of both the Kalman filter and the related Rauch-Tung-Striebel smoother [13]. This algorithm incorporates noisy measurements of a given multiscale process to calculate both smoothed estimates *and* the associated error covariances. Another algorithm that has been developed

is one to compute likelihoods [44]. Both of these algorithms have many attractive features, including high parallelizability, constant computational complexity per finest-scale data point, and completely natural handling of data at possibly several resolution scales. In light of our earlier discussion, these features are particularly noteworthy for 2-D imaging problems, as they represent rather significant advantages over other formalisms for modeling of random fields. With other formalisms, such as the MRF formalism, the computation of optimal estimates or likelihoods are quite complicated, having a per-pixel complexity that grows with image size; furthermore, the incorporation of multiresolution data is not particularly natural, and in fact exacerbates the problem with algorithmic complexity.

Complementary to the efficient algorithms that the framework admits, is the richness of the class of phenomena that it captures. For instance, experimental results in [14] demonstrate that these models can be used to capture the statistical self-similarity exhibited by stochastic processes having generalized power spectra of the form $1/f^{\beta}$. Furthermore, in [45], the authors demonstrate that that all 1-D wide-sense Markov random processes and 2-D wide-sense Markov random fields (WSMRFs) can in principle be represented within the framework.

1.2 Thesis Contributions

The success that the multiscale framework has already enjoyed provides the motivation for the deeper investigations that are carried out in this thesis. As summarized in the following paragraphs, we explore both theory and applications.

1.2.1 A Theory for Multiscale Stochastic Realization

Just as Kalman filtering requires the prior specification of a state-space model, so do our multiscale estimation algorithms require such a prior specification. In this sense, systematic model-building tools provide a needed, important link between the fast estimation algorithms we have at our disposal and a host of practical applications. We build this needed link by developing a theory of *multiscale stochastic realization.* Given the second-order statistics of a zero-mean random process or field, our methods provide a systematic way to realize the given statistics, to any desired degree of fidelity, as the finest scale of a multiscale process.

A central component of our development of this modeling approach is an analysis of the nature of the information interface provided by multiscale process state. Just as the state in a Gauss-Markov representation of a time series acts an interface between the past and the future of the process, so the state in a multiscale process must act as an interface, only now that interface is among multiple subsets of the process, not just two. Nevertheless, the similarity between the time-series stochastic realization problem and the multiscale realization problem is great enough that some of the tools used to analyze the former can be used to analyze the latter: motivated by Akaike's use of canonical correlation analysis to develop both exact and reduced-order models for time series [2], we too harness this tool from multivariate statistics to develop our multiscale models, both full and reduced-order.

In the course of our analysis, we uncover an interesting and non-trivial difference between time-series stochastic processes and multiscale stochastic processes. With regard to the former, a well-known property is that under fairly general conditions, a vector-valued random process $y(t)$ evolving in time (either discrete or continuous) can be represented by minimal state-space realization in which the random driving term $w(t)$ is a linear function

of $y(\tau)$, $-\infty < \tau < \infty$ [41]. This type of realization is referred to as an *internal* stochastic realization, because everything internal to the state-space model (i.e., $x(t)$ and $w(t)$) is obtainable directly from the observed process $y(t)$. A standard example of an internal realization is the so-called *innovations* representation, in which the driving noise is the innovations process produced by either a forward-running or backwards-running Kalman filter associated with *any* state-space realization of the process [2,41,56]. In contrast to the time-series case, there sometimes exists in the multiscale context a so-called *external* realization (i.e., a realization that is not internal) having a lower dimension than any internal realization. Here we are generalizing the internal concept to the multiscale context to mean a realization having state vectors and driving noises that are all obtainable as functions of the observed finest-scale process.

While most of our efforts focus on the development of tools to build internal realizations, we also briefly consider external realizations. We develop a method for building these external realizations, and we compare and contrast the resulting models to our internal ones. Ultimately, we find that both have their strengths and areas of application, with neither being universally preferable.

1.2.2 A Multiresolution Approach to Discriminating Targets from Clutter in SAR Imagery

In most detection and estimation problems, there is no ready availability of a complete statistical description of the quantities relevant to the problem, and thus, in these cases, the model-building techniques just described are not directly applicable. Instead, we must build an appropriate multiscale model from the observed data directly. We consider an important problem in automatic target recognition (ATR), for which we must apply so-called techniques of *multiscale model identification.*

We consider ATR for the case of a system whose inputs are synthetic-aperture radar (SAR) images. Within this problem domain, we both develop and extensively test a new algorithm for discriminating man-made objects from natural clutter. The novel feature of our approach is its exploitation of the characteristically distinct variations in speckle pattern, for imagery of natural clutter and of man-made objects, as image resolution is varied from coarse to fine. The fact that speckle has multiresolution characteristics is also noted and exploited in [61]. However, in contrast to that work, where the different characteristics of natural clutter and man-made objects are used to analyze individual image pixels, in this paper we use our multiscale framework to model and exploit these characteristics over entire blocks of imagery.

Within our multiscale framework, we build a pair of models: one for SAR imagery of natural clutter and another for imagery of man-made objects. We then use these models to define a multiresolution discriminant as the likelihood ratio for distinguishing between the two image types, given a multiresolution sequence of images of a region of interest (ROI). As we will see, by using the multiresolution modeling framework, the calculation of the likelihoods needed for our discriminant is extremely simple.

We incorporate this likelihood ratio into an existing, established discriminator that was developed at Lincoln Laboratory as part of a complete ATR system [40,52]. To classify a given ROI, we merge the information provided by our likelihood ratio with the measured values of a small number of size and brightness features. We have applied the resulting, new discriminator to an extensive data set of 0.3-meter resolution, HH polarization imagery gathered with the Lincoln Laboratory millimeter-wave SAR. The detection results are ex-

tremely good. In particular, the new discriminator achieves a significant improvement in receiver operating characteristics, compared to an optimized version of the standard discriminator that is traditionally used in the Lincoln Laboratory ATR system. This result conclusively demonstrates that multiresolution methods have an effective and important role to play in SAR ATR algorithms.

1.2.3 Overcoming the Problem of Blockiness

In spite of the success of the multiscale framework with regard to computational efficiency, mean-square estimation error, and ability to supply error covariance information, the framework, as developed up to this point in time, has a characteristic that would appear to limit its utility in certain applications. Specifically, estimates based on the types of multiscale models previously proposed may exhibit a visually distracting blockiness [46].

We eliminate this blockiness by discarding the standard assumption that distinct nodes on a given level of the multiscale process must correspond to disjoint portions of the image domain; instead, we allow a correspondence to overlapping portions of the image domain, thereby eliminating the hard boundaries between pixels. We use these so-called overlapping-tree models for both modeling and estimation. In particular, we develop an efficient multiscale algorithm for generating sample paths of a random field whose second-order statistics match a prespecified covariance structure, to any desired degree of fidelity. Furthermore, we demonstrate that under easily satisfied conditions, we can "lift" a random field estimation problem to one defined on an overlapped tree, resulting in an estimation algorithm that is computationally efficient, directly produces estimation error covariances, and eliminates any blockiness in the reconstructed imagery without any sacrifice in the resolution of fine-scale detail.

1.3 Thesis Organization

Chapter 2 lays the foundation for our later developments by reviewing a number of established results in multiscale theory, Markov random field theory and canonical correlation analysis.

Chapters 3, 4 and 5 represent the core of our development. Chapter 3 describes our work in multiscale stochastic realization. In turn Chapter 4 describes our use of the multiscale framework to aid in discriminating targets from clutter in SAR imagery. Finally, Chapter 5 describes our development of overlapping-tree models for eliminating blocky artifacts.

Chapter 6 provides another look at our contributions, with a hindsight perspective, and contains some suggestions for future research directions. Finally, Appendices A-D augment the content of Chapters 2-5 with more detailed material that has been offset to avoid disruption of the main flow of ideas.

Chapter 2

Preliminaries

Before we begin in earnest our detailed development, we introduce here certain preliminary concepts and background material concerning stochastic processes and statistics. A significant portion of this material is related to the multiscale framework per se, but also included are elements of Markov random field theory and canonical correlation theory. Once this background material has been reviewed, we will be much better equipped to understand and explore the new developments of the subsequent chapters.

In outline, this review proceeds in the following way. We begin with a more formal introduction to the multiscale framework, including a compilation of useful notation, a description of the dynamics of these processes, and some insight into the information content of process state. We highlight the Markov property of multiscale stochastic processes, and describe the efficient algorithms to which this property leads.

Next we review the definition and properties of wide sense (WS) reciprocal processes and Markov random fields (MRFs). We focus primarily on wide-sense MRFs (WSMRFs) defined on discrete toroidal lattices, for which the computational efficiency of the FFT can be brought to bear. Then, borrowing from multivariate statistics, we review the needed elements of canonical correlation theory, including some computational issues.

Finally, we return to a consideration of multiscale processes. Specifically, we exemplify the richness of the multiscale model class by demonstrating that *all* WS Markov random processes and fields, indexed on discrete lattices, have multiscale representations.

As a final remark, the material here has been compiled from various sources, and with only a few exceptions, none of the results are new. The chapter's intended purpose is to assemble together in a single place the background material needed to make the overall thesis document self-contained.

2.1 Introduction to the Multiscale Framework

2.1.1 State-Space Models Indexed on Trees

The models of interest to us, and originally introduced in [13, 46], describe multiscale stochastic processes indexed by nodes on a *tree*. A q^{th}-order tree is a pyramidal structure of nodes connected such that each node has q offspring nodes. An example of a 2nd-order tree or *dyadic* tree is depicted in Figure 2-1. Each horizontal level can be interpreted as a distinct scale, with the scales progressing from coarse to fine as the tree is traversed from top to bottom. We index these scales as $0, 1, \ldots$, with the top level being scale 0, the next

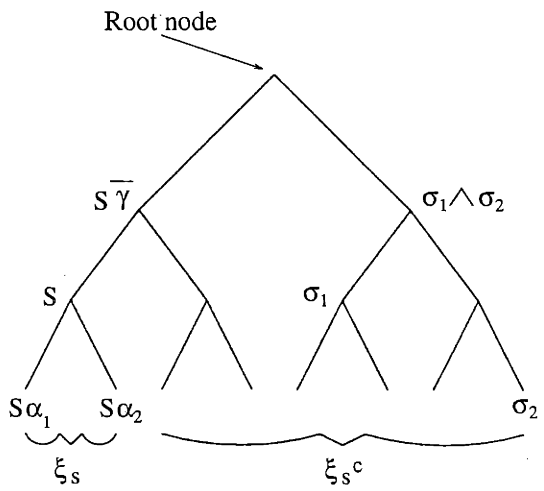


Figure 2-1: The first four levels of a dyadic tree are shown. The parent of node s is denoted by $s\bar{\gamma}$ and the two offspring are denoted by $s\alpha_1$ and $s\alpha_2$. The random vectors ξ_s and ξ_{s^c} contain, respectively, the finest-scale state information that does and does not descend from the node s.

level being scale 1, and so forth.

We denote nodes on the tree with an abstract index s, and we associate with each node a vector-valued state $x(s)$. In general, the q^m state vectors at the m-th level of the tree can be interpreted as "information" about the m-th scale of the process. To facilitate the description of traversal operations on the tree, we introduce shift operators, which play roles analogous to the forward and backward shift operators in discrete-time systems. In particular, we define an upward (fine-to-coarse) shift operator $\bar{\gamma}$ such that $s\bar{\gamma}$ is the *parent* of node s. We also define a corresponding set of downward (coarse-to-fine) shift operators α_i, $i = 1, 2, ..., q$ such that the q *offspring* of node s are given by $s\alpha_1$, $s\alpha_2$..., $s\alpha_q$. We let $m(s)$ denote the level of node s, so that, for example, $m(s\bar{\gamma}) = m(s) - 1$ and $m(s\alpha_i) = m(s) + 1$. Finally, we introduce the $\wedge$ operator, defined such that $s \wedge \sigma$ is the ancestor of both s and σ that is closest to the finest scale. Figure 2-1 depicts an example of the relative locations of $s, s\bar{\gamma}$, and $s\alpha_1, s\alpha_2$ in a dyadic tree; also depicted are the relative locations of σ_1, σ_2 and $\sigma_1 \wedge \sigma_2$.

The dynamics that implicitly provide a statistical characterization of $x(s)$ are given by

$$x(s) \;=\; A(s)x(s\bar{\gamma}) \;+\; B(s)w(s). \tag{2.1}$$

In this equation, $A(s)$ and $B(s)$ are matrices of appropriate size. The recursion is initialized at the root node, $s = 0$, with a state variable $x(0)$, which is assumed throughout this thesis to be zero mean. The term $w(s)$ represents white driving noise, which is both zero mean and uncorrelated with the initial condition $x(0)$. If we interpret each level as a representation of one scale of the process, then we see that (2.1) describes the evolution of a process from coarse to fine scales. The term $A(s)x(s\bar{\gamma})$ represents interpolation or prediction down to the next level and $B(s)w(s)$ represents new information or detail added as the process evolves from one scale to the next.

We emphasize that our notion of $x(s)$ representing scale "information" is deliberately abstract. The specific details of what $x(s)$ represents are dependent on a multitude of considerations about the particular application at hand. Dominant among these considerations is the statistical structure of the particular underlying stochastic phenomenon, but

also important is the tradeoff between model simplicity and model accuracy. When all of these considerations have been appropriately weighed, there are many possible outcomes for what $x(s)$ should represent. For instance, values of the process at level m may correspond to averages of the offspring values at level $m+1$; in this case, if $q = 2$, then the values $x(s)$ can be interpreted as *scaling* coefficients in a Haar wavelet representation of the finest scale process [14]. Alternatively, the values at different levels may correspond to decimated versions of the finest scale process [45]. These are only two of many possibilities; our stochastic realization theory in Chapter 3 and our SAR modeling work in Chapter 4 will provide others.

2.1.2 Characterization of First-order and Second-order Statistics

In a straightforward manner, we can determine the first-order and second-order statistics of the process $x(s)$. With regard to the first-order statistics, since both $x(0)$ and $w(s)$ are zero mean, it follows trivially from (2.1) that $x(s)$ is zero mean. Turning to second-order statistics, we denote the covariance of $x(s)$ by $P_{x(s)}$ and the cross-covariance between $x(s)$ and $x(\sigma)$ by $P_{x(s)x(\sigma)}$,

$$\begin{aligned} P_{x(s)} &\equiv E\left[x(s)x^T(s)\right], \\ P_{x(s)x(\sigma)} &\equiv E\left[x(s)x^T(\sigma)\right]. \end{aligned}$$

The covariance $P_{x(s)}$ evolves according to a Lyapunov equation on the tree:

$$P_{x(s)} = A(s)P_{x(s\bar{\gamma})}A^T(s) + B(s)B^T(s). \tag{2.2}$$

To relate the cross-covariance to the covariance, we introduce the state-transition matrix $\Phi(s,\sigma)$, defined via the following recursive relationship:

$$\Phi(s,\sigma) \equiv \begin{cases} I & \text{if } s = \sigma \\ A(s)\Phi(s\bar{\gamma},\sigma) & \text{if } m(s) > m(\sigma). \end{cases}$$

Finally, we have that

$$P_{x(s)x(\sigma)} = \Phi(s, s\wedge\sigma)P_{x(s\wedge\sigma)}\Phi^T(\sigma, s\wedge\sigma).$$

2.1.3 Relation to Gauss-Markov Time Series Models

We can obtain further insight into the structure of the multiscale model class described by (2.1) by relating it to a more traditional class of stochastic processes, namely Gauss-Markov time series models. As discussed in Chapter 1, the dynamics of these time series are described by the state-space equation

$$\begin{aligned} z(n+1) &= A(n)z(n) + w(n) \\ y(n) &= C(n)z(n), \end{aligned}$$

where $z(n)$ denotes the state of the process at time n, $A(n)$ is the one-step transition matrix, $C(n)$ is the observation matrix, and $w(n)$ is zero-mean, white noise driving term.

To bring out clearly the connection between Gauss-Markov time series models and our multiscale models, we view the set of integers as a first order tree, in which n is connected

to $n-1$. In this sense, the multiscale model class is a clear intellectual descendant of the Gauss-Markov model class, with the modifications that the dynamics are indexed by scale rather time, and they progress on higher-order trees.

2.1.4 Markov Property of Multiscale Processes

Multiscale processes possess an important Markov property, stemming directly from the whiteness of the driving term $w(s)$ in the recursion in (2.1). This property is not only essential to the extremely efficient, highly parallelizable algorithms that the multiscale framework admits, but will also be fundamental to our approach to stochastic realization (Chapter 3) and to our analysis motivating the need for overlapping trees (Chapter 5).

The most obvious form of Markovianity possessed by (2.1) is the Markovianity in scale, as scale progresses from coarse to fine. This property is readily discernible, and is actually subsumed by a more general form of Markovianity. To see this more general Markov property, we first note that any given node on a q^{th}-order tree can be viewed as a boundary between $q+1$ subsets of nodes, where q of these subsets correspond to paths leading towards offspring and one corresponds to a path leading towards the parent.[1] With this boundary notion in mind, the Markov property can be stated as follows: conditioned on the value of the state at any node, the values of the states in the corresponding $q+1$ subsets of nodes extending away from s are uncorrelated.

In much of our work, attention will be focused on the finest scale of multiscale processes. To relate the Markov property to this finest scale, we introduce some special notation. We associate with each tree node s a set $\mathcal{F}_s$, where $\mathcal{F}_s$ contains all of the finest-scale nodes that descend from s:

$$\mathcal{F}_s \;=\; \{\sigma;\ \sigma \text{ is a descendant of } s, \text{ and is at the finest scale}\},$$

We also associate with each node s the random vectors ξ_s and ξ_{s^c}. These vectors contain, respectively, the finest-scale state information that does and does not descend from s. More specifically, ξ_s contains the $|\mathcal{F}_s|$ elements of the set $\{x(\sigma);\ \sigma \in \mathcal{F}_s\}$, stacked into a vector, while ξ_{s^c} contains the $(|\mathcal{F}_0| - |\mathcal{F}_s|)$ elements of the set $\{x(\sigma);\ \sigma \in \mathcal{F}_0\}$ that are not in the set $\{x(\sigma);\ \sigma \in \mathcal{F}_s\}$. These conventions are illustrated in Figure 2-1; as a special case, we note that ξ_0 comprises the entire finest-scale process.

For any given multiscale process, there is a very precise relationship between $x(s)$ and ξ_σ. To capture this relationship, we introduce the matrix $H_{\sigma|s}$ and the random vector $\tilde{\xi}_{\sigma|s}$. The matrix $H_{\sigma|s}$ is implicitly defined by the relation

$$E\left(\xi_\sigma | x(s)\right) \;=\; H_{\sigma|s}x(s).$$

In turn, the random vector $\tilde{\xi}_{\sigma|s}$ is defined to be the residual in the least-squares estimate of ξ_σ with $x(s)$:

$$\begin{aligned}\tilde{\xi}_{\sigma|s} &\equiv \xi_\sigma - E\left(\xi_\sigma | x(s)\right)\\ &= \xi_\sigma - H_{\sigma|s}x(s).\end{aligned}$$

The Markov property, as it relates explicitly to the finest scale, can now be stated as

[1] The root node is an exception, having only q offspring and no parent. Also, the finest-scale nodes are exceptions, each having a single parent, but no offspring.

follows:

$$\tilde{\xi}_{s\alpha_i|s} \perp \tilde{\xi}_{(s\alpha_i)^c|s}, \qquad i = 1, 2, \ldots q. \tag{2.3}$$

The uncorrelatedness stipulated in (2.3) is extremely important, and as we will see in Chapter 3, will be the guiding relation in our procedure for building multiscale models.

Since

$$E\left[\xi_{s\alpha_i} \mid x(s)\right] = H_{s\alpha_i|s}x(s) \quad \text{and} \quad E\left[\xi_{s^c} \mid x(s)\right] = H_{s^c|s}x(s),$$

we can express (2.3) in the following equivalent way:

$$\xi_0 = \begin{pmatrix} \xi_{s\alpha_1} \\ \xi_{s\alpha_2} \\ \vdots \\ \xi_{s\alpha_q} \\ \xi_{s^c} \end{pmatrix} = \begin{pmatrix} H_{s\alpha_1|s} \\ H_{s\alpha_2|s} \\ \vdots \\ H_{s\alpha_q|s} \\ H_{s^c|s} \end{pmatrix} x(s) + \begin{pmatrix} \tilde{\xi}_{s\alpha_i|s} \\ \tilde{\xi}_{s\alpha_2|s} \\ \vdots \\ \tilde{\xi}_{s\alpha_q|s} \\ \tilde{\xi}_{s^c|s} \end{pmatrix}, \tag{2.4}$$

with

$$x(s),\ \tilde{\xi}_{s\alpha_1|s}, \tilde{\xi}_{s\alpha_2|s}, \ldots, \tilde{\xi}_{s\alpha_q|s}, \text{ and } \tilde{\xi}_{s^c} \text{ uncorrelated.}$$

This form is useful for discerning the relationship between the dimension of $x(s)$ and the correlation among the vectors $\xi_{s\alpha_1}$, $\xi_{s\alpha_2}, \ldots,$ $\xi_{s\alpha_q}$ and ξ_{s^c}. In particular, (2.4), together with the uncorrelatedness of the terms $\tilde{\xi}_{s\alpha_i}$ and $\tilde{\xi}_{s^c}$ implies that for $i \neq j$,

$$E\left(\xi_{s\alpha_i}\xi_{s\alpha_j}^T\right) = H_{s\alpha_i|s}P_{x(s)}H_{s\alpha_j|s}^T \quad \text{and} \quad E\left(\xi_{s\alpha_i}\xi_{s^c}^T\right) = H_{s\alpha_i|s}P_{x(s)}H_{s^c|s}^T \tag{2.5}$$

By elementary linear algebra [60], the rank of this cross-covariance is upper-bounded by the rank of $P_{x(s)}$, which in turn is upper-bounded by the dimension of $x(s)$. We have thus proved the following Proposition.

Proposition 1 *Corresponding to any finest-scale correlation structure, there is a lower-bound on the dimension required for each state $x(s)$ in an exact realization:*

$$\textit{dimension}(x(s)) \geq \max_{i \neq j} \left\{ \textit{rank}\left[E\left(\xi_{s\alpha_i}\xi_{s\alpha_j}^T\right)\right]\right\},$$

and

$$\textit{dimension}(x(s)) \geq \max_{i} \left\{ \textit{rank}\left[E\left(\xi_{s\alpha_i}\xi_{s^c}^T\right)\right]\right\}.$$

This proposition provides some insight into the multiscale stochastic realization problem. To see this fact, let us consider the problem of building a multiscale process indexed on a dyadic tree, such that the finest-scale process has exactly the following covariance:

$$E\left[\begin{pmatrix} \xi_{0\alpha_1} \\ \xi_{0\alpha_2} \end{pmatrix} \begin{pmatrix} \xi_{0\alpha_1}^T & \xi_{0\alpha_2}^T \end{pmatrix}\right] = \begin{pmatrix} P_1 & P_{12} \\ P_{12}^T & P_2 \end{pmatrix}.$$

The structure of our multiscale stochastic processes is such that the root node is the only location in the tree that we can inject information common to both $\xi_{0\alpha_1}$ and $\xi_{0\alpha_2}$. Thus, we would intuitively expect a close relationship between the dimension of the root node and the correlation between $\xi_{0\alpha_1}$ and $\xi_{0\alpha_2}$. In fact, as shown by Proposition 1, an exact realization of the desired covariance requires a root node state of dimension at least as large as the rank of P_{12}. This dimension constraint is a rather stringent, especially in the (rather likely) case that P_{12} will have full rank. There is thus a clear need for reduced-order, approximate realizations, as we will discuss at length in Chapter 3.

2.1.5 Signal Processing in the Multiscale Framework

One of the primary reasons the multiscale framework is useful is that it admits extremely efficient, highly parallelizable algorithms that allow one to incorporate noisy measurements $y(s)$ of a given multiscale process $x(s)$ to calculate (i) the smoothed estimate of $x(s)$ [13–15] and (ii) the likelihood of $y(s)$ [43, 44]. For the purposes of these algorithms, the noisy measurements $y(s)$ are modeled as

$$y(s) = C(s)x(s) + v(s).$$

In this equation, $C(s)$ is a matrix specifying the nature of the process observations as a function of both spatial location and scale. The term $v(s)$ represents additive white noise that corrupts the observations. In this manner, we can accommodate observations that are arbitrarily distributed in space and scale.

The particular algorithm that will be of interest to us is the one that carries out multiscale-based estimation. This algorithm allows us to compute the linear least-squares estimate[2] $\hat{x}(s)$,

$$\hat{x}(s) \equiv E[x(s) \mid y(\sigma), \sigma \in \mathcal{M}],$$

based on noisy observations $\{y(\sigma);\ \sigma \in \mathcal{M}\}$, where $\mathcal{M}$ is the arbitrary set of nodes for which we have observations. The algorithm also computes the associated error covariance $P_{\tilde{x}(s)}$,

$$\begin{aligned} P_{\tilde{x}(s)} &\equiv E\left[\tilde{x}(s)\tilde{x}^T(s)\right], \\ \tilde{x}(s) &\equiv x(s) - \hat{x}(s). \end{aligned}$$

As discussed in detail in [13,46], this algorithm takes explicit advantage of the Markovian structure of $x(s)$ on the tree. Specifically, the algorithm incorporates the measurements into the estimates via two recursive sweeps, with each sweep following the structure of the tree. The first sweep proceeds from fine to coarse scales, calculating at each node s the best estimate (and associated error covariance) of $x(s)$ given data in the subtree below node s. In turn, the second sweep proceeds from coarse to fine scales, calculating at each node s the best estimate $\hat{x}(s)$ (and associated error covariance $P_{\tilde{x}(s)}$) of $x(s)$ given *all* of the data.

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

2.1.6 Computational Complexity of Multiscale Processing

Just as with the more traditional Gauss-Markov time-series models, there is a close relation between the complexity of our multiscale models and the speed of the algorithms associated with these models. To obtain insight into these relationships, we now develop explicit expressions for the computational complexity of both our algorithm for multiscale-based estimation and our algorithm for simulating the recursion in (2.1). These expressions will highlight one of the great strengths of the multiscale framework, and will also point to some of the challenges we must meet.

There are three multiscale model parameters of fundamental interest in our discussion: (i) the number K of pixels in the image domain, (ii) the number N of finest-scale nodes in the multiscale model, and (iii) the maximal dimension λ of any state vector $x(s)$ in the multiscale model. The first two of these parameters are closely related; in fact, in previous applications, N has been identical to K. On the other hand, in anticipation of developments in Chapter 5 of this thesis, where we develop estimation with overlapping trees, we also allow for values of N that are greater than K; we relate the two by $K = rN$, where $0 < r < 1$ is a measure of the degree of overlap, with smaller r corresponding both to more overlap.

The two-sweep structure of our estimation algorithm implies that each node of the tree is visited exactly twice, where the computations at each node involve a number of floating point operations proportional to the cube of the state vector at the given node. Thus, since the total number of nodes satisfies the bounds

$$N \ < \ \text{total number of nodes} \ < \ \frac{q}{q-1}N,$$

we conclude that application of the estimation algorithm requires a total of $\mathcal{O}(\lambda^3 N)$ floating point operations. Similarly, the simulation of the coarse-to-fine recursion in (2.1) requires a total of $\mathcal{O}(\lambda^2 N)$ floating point operations.[3]

The foregoing complexity figures imply that a serial implementation requires a total computational time per image pixel of $\mathcal{O}(\lambda^3/r)$ for estimation and $(\mathcal{O}(\lambda^2/r))$ for simulation. However, we are not constrained to use a serial implementation: any sweep over the tree can be carried out with all calculations at each tree level being performed in parallel. If maximal advantage is taken of this parallelism, then the computational time per image pixel becomes

$$O\left(\frac{\lambda^3 \log(K/r)}{K}\right)$$

for estimation and

$$O\left(\frac{\lambda^2 \log(K/r)}{K}\right)$$

for simulation, both of which actually *decrease* as K increases. The point here is that we can achieve dramatic computational benefit as long as the maximal dimension n of the state model and the amount of overlap (as measured by $1/r$) are not too large. We will see in subsequent chapters that our procedures allow us to meet these criteria.

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

2.2 Wide-Sense Markov Random Fields

Markov random fields (MRFs), which were discussed at a high level in Chapter 1, have been used to guide decisionmaking in a wide variety of statistical signal processing contexts, including image restoration, image segmentation and anomaly detection. Part of the appeal of these models is that they are well suited for capturing the spatial continuity that is characteristic of many images; they allow one to directly control the statistical relationship between a pixel value and the values of a relatively small number of neighboring pixels. We too will have occasion to use MRFs, and for this purpose we here summarize some of their properties. This material has been assembled from a variety of sources, including [11, 18] and [65].

2.2.1 Definition

We are particularly interested in *wide-sense* reciprocal processes and Markov random fields, defined on discrete lattices in 1-D and 2-D, respectively. A 1-D stochastic process $z(i)$, $i \in \mathcal{Z}$ (where $\mathcal{Z}$ is the set of integers) is said to be a *wide-sense reciprocal process* if and only if the linear least-squares estimate of $z(i)$, given values of the process at all other points, depends only on the values in some neighborhood of points around i. More precisely, z_i is a WS reciprocal process if and only if

$$E\left(z(i) \mid z(i-j),\ j \neq 0\right) \;=\; E\left(z(i) \mid z(i-j),\ j \in D\right), \tag{2.6}$$

where D denotes the set of neighbor offsets, such that $i-j$ is a neighbor of i if and only if $j \in D$. We analogously define a 2-D stochastic process $z(i,j)$, $(i,j) \in \mathcal{Z} \times \mathcal{Z}$ to be a *wide-sense Markov random field* if and only if

$$E\left(z(i,j) \mid z(i-k,j-l),\ (k,l) \neq (0,0)\right) \;=\; E\left(z(i,j) \mid z(i-k,j-l),\ (k,l) \in D\right) \tag{2.7}$$

where once again D denotes the set of neighbor offsets.

From these definitions, we see that an essential component of the specification of a particular reciprocal process or MRF is the specification of its neighborhood system. For 1-D applications, the neighborhood has the following simple structure:

$$D \;=\; \{-R, -(R-1), \ldots, -1, 1, \ldots, R-1, R\},$$

where we refer to R as the *order* of the neighborhood. For 2-D applications, there is a standard hierarchical sequence of neighborhoods D [12]; this sequence is illustrated in Figure 2-2 for neighborhoods up to order seven.

2.2.2 Autoregressive Representation

By combining (2.6) and (2.7) with the orthogonality principle of linear least-squares estimation, it follows that both WS reciprocal processes and MRFs must satisfy an autoregressive difference equation. Focusing on the 2-D case, this difference equation has the form

$$z(i,j) \;=\; \sum_{(k,l)\in D} r(k,l)z(i-k,j-l) \;+\; w(i,j), \tag{2.8}$$

```
          7 6 7
        5 4 3 4 5
      7 4 2 1 2 4 7
      6 3 1 t 1 3 6
      7 4 2 1 2 4 7
        5 4 3 4 5
          7 6 7
```

Figure 2-2: The neighborhoods of lattice site t for orders one through seven. For example, the first-order neighborhood consists of just the two vertical and two horizontal nearest neighbors.

where the driving noise $w(i,j)$ is correlated with the signal $z(i-k,j-l)$ in the following way:

$$E\left(w(i,j)z(i-k,j-l)\right) = \begin{cases} \sigma_{i,j}^2 & (k,l)=(0,0) \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

This autoregressive relation can be applied to a finite lattice (of dimension, say, $N \times N$) by treating the given finite lattice as *toroidal*, so that $z(i,j) = z(i \bmod N, j \bmod N)$ and similarly $w(i,j) = w(i \bmod N, j \bmod N)$. So, for example, the first-order neighbors of the lattice point $(0,0)$ are the four lattice points $(0,1)$, $(1,0)$, $(0,N-1)$, and $(N-1,0)$.

To be consistent with (2.8) and (2.9), the driving noise must be spatially colored, having a correlation structure that is closely related to the autoregressive weights in (2.8):

$$E\left(w(i,j)w(i-k,j-l)\right) = \begin{cases} \sigma_{i,j}^2 & (k,l)=(0,0) \\ -\sigma_{i,j}^2 r(k,l) & (k,l)\in D \\ 0 & \text{otherwise.} \end{cases} \tag{2.10}$$

We assume throughout the rest of our development that $\sigma_{i,j}^2$ has a constant value, independent of position (i,j). Thus, to be consistent with (2.10), the autoregressive weights $r(k,l)$ must be symmetric, in the sense that $r(k,l) = r(-k,-l)$.

Interestingly, our original definition (2.7) is not only sufficient for (2.8) and (2.9) to hold, it is also necessary. In other words, we can take (2.8) and (2.9) as the definition of a WSMRF, where this alternative definition is equivalent with (2.7). We have already considered the sufficiency of (2.8) and (2.9); necessity is easily established as follows:

$$\begin{aligned} E\left(z(i,j) \mid z(i-k,j-l),(k,l)\neq(0,0)\right) & \\ &= E\left(z(i,j) \mid z(i-k,j-l), w(i-k,j-l),(k,l)\neq(0,0)\right) \\ &= E\left(z(i,j) \mid z(i-k,j-l), w(i-m,j-n),(k,l)\in D,\right. \\ &\qquad\qquad \left.(m,n)\neq(0,0)\right) \\ &= E\left(z(i,j) \mid z(i-k,j-l),(k,l)\in D\right). \end{aligned}$$

In the first and third equalities, we have used (2.9), while in the second equality, we have used (2.8).

We can obtain additional insight by recasting (2.8) and (2.10) in matrix-vector form.

We let z and w denote, respectively, the random field and driving noise stacked into lexicographically ordered vectors:

$$\begin{aligned} z &\equiv \Big(z(0,0) \quad z(1,0) \quad \dots \quad z(N-1,0) \quad z(0,1) \quad \dots \quad z(N-1,N-1) \Big)^T, \\ w &\equiv \Big(w(0,0) \quad w(1,0) \quad \dots \quad w(N-1,0) \quad w(0,1) \quad \dots \quad w(N-1,N-1) \Big)^T \end{aligned} \tag{2.11}$$

We denote the covariance of z by P_z and that of w by P_w. In keeping with (2.8), the vectors z and w are linearly related,

$$Gz = w, \tag{2.12}$$

where the $N^2 \times N^2$ matrix G is symmetric, positive definite and block circulant, with circulant blocks. In particular,

$$\begin{aligned} G &= \text{circulant}(G_0, G_1, \dots, G_{N-1}) \\ &= \begin{pmatrix} G_0 & G_1 & \cdots & G_{N-1} \\ G_{N-1} & G_0 & \cdots & G_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ G_1 & G_2 & \cdots & G_0 \end{pmatrix}, \end{aligned}$$

with $G_j = G_{N-j}$, for $j = 1, 2, \dots, N-1$, and where each block-component G_i is $N \times N$, symmetric and circulant:[4]

$$\begin{aligned} G_i &= \text{circulant}(g_{i,0}, g_{i,1}, \dots, g_{i,N-1}), \\ g_{i,j} &= \begin{cases} 1 & i = j = 0 \\ -r(i_1, j_1) & (i_1, j_1) \in D, \text{ with } \begin{cases} i_1 = \frac{-N}{2} + \left(\left(i + \frac{N}{2} \right) \bmod \frac{N}{2} \right) \\ j_1 = \frac{-N}{2} + \left(\left(j + \frac{N}{2} \right) \bmod \frac{N}{2} \right) \end{cases} \\ 0 & \text{otherwise.} \end{cases} \end{aligned},$$

Not only does G serve to relate w and z, as in (2.12), but also, from (2.10) and (2.11), we see that the covariance of w satisfies

$$P_w = G,$$

and hence, applying (2.12), we arrive at the somewhat surprising result that the covariance of z satisfies

$$P_z^{-1} = G.$$

This relation highlights the close relationship between a model (i.e., the matrix G, which captures the weights of an autoregressive representation of the field) and the inverse of the field covariance.

[4]For simplicity, we assume here that N is even.

2.2.3 The FFT and its Relation to WSMRFs Indexed on Toroidal Lattices

The eigenstructure of both G and its inverse P_z is related to the FFT in such a way that we can exploit the FFT's computational efficiency to carry out a number of signal processing tasks: (i) calculation of the correlation function $R_{zz}(k,l) \equiv E\left(z(i,j)z(i-k,j-l)\right)$, (ii) generation of random field sample paths, (iii) least-squares estimation, (iv) spectral estimation. In the course of this thesis, we will do all of these things, and so here we summarize the role played by the FFT.

Eigenanalysis

To clarify the relation between the FFT and G, we define the complex scalar θ_k via

$$\theta_k \equiv \exp\left(\sqrt{-1}\frac{2\pi}{N}k\right),$$

which we use to define both the complex N-vector t_i,

$$t_i \equiv \left(\begin{array}{cccc} \theta_i^0 & \theta_i^1 & \cdots & \theta_i^{N-1} \end{array}\right)^T,$$

and the complex N^2-vector $f_{i,k}$,

$$f_{i,k} \equiv \left(\begin{array}{cccc} \theta_k^0 t_i^T & \theta_k^1 t_i^T & \cdots & \theta_k^{N-1} t_i^T \end{array}\right)^T.$$

This vector $f_{i,k}$ is relevant to our discussion for two reasons. First, the inner-product $f_{i,k}^H z$ is exactly the (i,k)-th discrete Fourier coefficient in the transform of the field:

$$f_{i,k}^H z = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \theta_m^{-i}\theta_n^{-k} z_{m,n}. \tag{2.13}$$

Second, and more importantly, $f_{i,k}$ is an eigenvector of any $N^2 \times N^2$ symmetric, block circulant matrix, having N circulant blocks, and so, $f_{i,k}$ is an eigenvector of both P_z and G, for $i,k = 0,1,\ldots,N-1$. We thus consolidate all of the eigenvectors of H into the columns of a matrix F,

$$F = \left(\begin{array}{ccccccc} f_{0,0} & f_{1,0} & \cdots & f_{N-1,0} & f_{0,1} & \cdots & f_{N-1,N-1} \end{array}\right),$$

so that $F^H x$ is the discrete Fourier transform of x and F^{-H} is the inverse discrete Fourier transform, with

$$F^{-H} = \frac{1}{N^2}F.$$

Furthermore, we have that

$$\begin{aligned} P_z &= F\Lambda F^{-1} \\ &= F^{-H}\Lambda F^H, \end{aligned} \tag{2.14}$$

and

$$\begin{aligned} G &= F\Lambda^{-1}F^{-1} \\ &= F^{-H}\Lambda^{-1}F^{H}, \end{aligned} \tag{2.15}$$

where

$$\Lambda = \mathrm{diag}(\lambda_{0,0}, \lambda_{1,0}, \ldots, \lambda_{N-1,N-1}),$$

Applications of the FFT

Calculation of eigenvalues of P_z The eigenvalues of P_z and G are reciprocals of one another, and so if we can find the eigenvalues of either, then we can trivially obtain the eigenvalues of the other. Since we directly have available G, we calculate its eigenvalues first. This calculation can be carried out by simply taking the 2-D FFT of its first column. To see this fact, we let e_1 and s be column vectors, having the same dimension as z, with

$$\begin{aligned} e_1 &\equiv \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^T \\ s &\equiv \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}^T. \end{aligned}$$

Also, we let μ be a column vector defined as

$$\mu \equiv \Lambda^{-1}s, \tag{2.16}$$

so that its i-th component is equal to the i-th diagonal entry in Λ^{-1}. With these conventions established, we come to the key result, which is that

$$\begin{aligned} F^H(Ge_1) &= \Lambda^{-1}F^H e_1 \\ &= \Lambda^{-1}s \\ &= \mu, \end{aligned}$$

where the first line follows from (2.15), the second line from the definitions of F and s, and the third line from (2.16).

Calculation of $R_{zz}(\cdot,\cdot)$ Stationarity implies that there is a great deal of redundancy in the covariance matrix P_z, and in fact, we can extract the *entire* correlation structure $R_{zz}(\cdot,\cdot)$ from first column of P_z alone. To see this fact, let us stack the values of the correlation function $R_{zz}(\cdot,\cdot)$ into a vector r, defined as follows:

$$\begin{aligned} r &\equiv P_z e_1 \\ &= \begin{pmatrix} R_{zz}(0,0) & R_{zz}(1,0) & \cdots & R_{zz}(N-1,0) & R_{zz}(0,1) & \cdots \\ & & & & & R_{zz}(N-1,N-1) \end{pmatrix}^T. \end{aligned} \tag{2.17}$$

To calculate r, we first form λ, defined to be the column vector

$$\lambda \equiv \Lambda s \tag{2.18}$$

so that its i-th component is equal to the i-th diagonal entry in Λ. It then follows that

$$\begin{aligned} r &= P_z e_1 \\ &= F^{-H} \Lambda F^H e_1 \\ &= F^{-H} \Lambda s \\ &= F^{-H} \lambda, \end{aligned}$$

where the first line is a consequence of (2.14), the second line a consequence of the definitions of F and s, and the third line a consequence of (2.18). Hence, we can calculate efficiently the first column of P_z by computing the 2-D inverse-FFT of λ, which itself can be found by taking the reciprocals of the components of the 2-D FFT of the first column of G.

Generation of field sample paths To generate a sample path of z, we first generate a sample path of a zero-mean random vector ω, having identity covariance, and then we let

$$\begin{aligned} z &= P_z^{1/2} \omega \\ &= F^{-H} \Lambda^{1/2} F^H \omega. \end{aligned}$$

Thus, we can simulate z by the following four-step procedure: (i) generate a white field ω, (ii) apply a 2-D FFT to ω, (iii) filter the transformed field with $\Lambda^{1/2}$, and (iv) apply an 2-D inverse-FFT to the transformed, filtered field.

Least-squares estimation Let us consider now the calculation of the linear, least-squares estimate of z, given $y = z + v$, where v is a zero-mean random vector having covariance $\sigma_v^2 I$, for some positive scalar σ_v^2. We have the following sequence of identities:

$$\begin{aligned} E(z \mid y) &= P_z (P_z + \sigma_v^2 I)^{-1} y \\ &= F^{-H} \Lambda F^H \left[F^{-H} \left(\Lambda + \sigma_v^2 I \right) F^H \right]^{-1} y \\ &= F^{-H} \Lambda \left(\Lambda + \sigma_v^2 I \right)^{-1} F^H y, \end{aligned}$$

where the first line is a standard result in linear least-squares estimation, the second line follows from (2.14) and the third line follows by simple algebraic cancellation. Thus we can estimate z by the following three-step procedure: (i) apply a 2-D FFT to y, (ii) filter the transformed field with $\Lambda(\Lambda + \sigma_v^2 I)^{-1}$, and (iii) apply a 2-D inverse-FFT to the transformed, filtered field.

The estimation error covariance is

$$\begin{aligned} E((z - E(z|y))(z - E(z|y))^T) &= P_z - P_z \left(P_z + \sigma_v^2 I \right)^{-1} P_z \\ &= F \left(\Lambda - \Lambda^2 \left(\Lambda + \sigma_v^2 I \right)^{-1} \right) F^{-1}, \end{aligned} \tag{2.19}$$

where the second line reveals that the eigenvalues of the error covariance are given as follows:

$$\lambda_{i,j} - \frac{\lambda_{i,j}^2}{\lambda_{i,j} + \sigma_v^2}, \quad i,j = 0, 1, \ldots, N-1.$$

Since these eigenvalues are readily computed, it follows that the mean-square error (mse) of

the least-squares estimator is also readily computed. To see this latter fact, we note that by symmetry the diagonal elements of the error covariance all have the same value. But this common value is obviously equal to the arithmetic mean of the trace of the error covariance, which in turn is equal to the arithmetic mean of the eigenvalues of the error covariance. Hence, we conclude that

$$\text{mse} \;=\; \frac{1}{N^2}\sum_{i,j}\left(\lambda_{i,j} - \frac{\lambda_{i,j}^2}{\lambda_{i,j}+\sigma_v^2}\right).$$

Let us consider now the use of Monte-Carlo simulation to *estimate* the mse of the least-squares estimator. To carry out the simulation, we generate N_{samp} sample paths $z_1, z_2, \ldots, z_{N_{samp}}$ of the random field, and N_{samp} sample paths $v_1, v_2, \ldots, v_{N_{samp}}$ of the noise field, which we combine to create N_{samp} sample paths $y_1, y_2, \ldots, y_{N_{samp}}$ of the observed field, with $y_i = z_i + v_i$, for $i = 1, 2, \ldots, N_{samp}$. We then define e_i to be the error field $e_i = z_i - E(z_i \mid y_i)$ at the i-th trial. Finally, we estimate mse with the following unbiased estimator:

$$\hat{\text{mse}} \;\equiv\; \frac{1}{N_{samp}}\sum_{i=1}^{N_{samp}}\left(\frac{1}{N^2}e_i^T e_i\right).$$

Assuming that z_i and e_i are Gaussian, we find that the estimation error variance for $\hat{\text{mse}}$ is given as follows:

$$\begin{aligned}\text{Var}\,(\hat{\text{mse}} - \text{mse}) \;&=\; \frac{\text{Var}\left(e_i^T e_i\right)}{N_{samp}N^4} \\ &=\; \frac{2}{N_{samp}N^4}\sum_{k,l}\left(\lambda_{k,l} - \frac{\lambda_{k,l}^2}{\lambda_{k,l}+\sigma_v^2}\right)^2, \qquad (2.20)\end{aligned}$$

where in the second line we have used the fact that for any Gaussian random vector η having $\phi_1, \phi_2, \ldots$ as the eigenvalues of its covariance,

$$\text{Var}(\eta_i^T \eta_i) \;=\; 2\sum_i \phi_i^2.$$

The relation in (2.20) is quite useful for determining the number of trials N_{samp} required to increase our confidence in the estimate $\hat{\text{mse}}$ to any desired level. To be specific, by the central limit theorem, $\hat{\text{mse}}$ will be approximately Gaussian, and thus, for any tolerance ϵ and probability α, we can assert that

$$\text{Prob}\,(|\hat{\text{mse}} - \text{mse}| \;\leq\; \epsilon) \;\geq\; \alpha,$$

if the number of trials N_{samp} satisfies

$$N_{samp} \;\geq\; \frac{2\beta^2}{\epsilon^2 N^4}\sum_{k,l}\left(\lambda_{k,l} - \frac{\lambda_{k,l}^2}{\lambda_{k,l}+\sigma_v^2}\right),$$

where

$$\beta \equiv \Phi^{-1}\left(\frac{1+\alpha}{2}\right),$$

with

$$\Phi(x) \equiv \int_{-\infty}^{x} \frac{\exp(-t^2/2)}{\sqrt{2\pi}}\, dt.$$

Estimation of $R_{zz}(\cdot,\cdot)$ Suppose we have N_{samp} sample paths $z_1, z_2, \ldots, z_{N_{samp}}$ of some zero-mean, stationary WSMRF, whose correlation vector r we wish to estimate. Exploiting the linear relationship (2.17) between r and λ, we proceed by first estimating λ, thus yielding $\hat{\lambda}$ and then letting

$$\hat{r} = F^{-H}\hat{\lambda}. \tag{2.21}$$

To estimate λ, we use the standard technique of periodogram averaging. In particular, we define ω_i as

$$\omega_i \equiv F^H z_i,$$

for which $E(\omega_i \omega_i^H) = N^2\Lambda$. Then, letting $(\omega_i)_{j,k}$ denote the (j,k)-th component of ω_i, we use the following unbiased estimate for the j-th component of $\hat{\lambda}$:

$$\hat{\lambda}_{j,k} = \frac{1}{N_{samp}N^2}\sum_{i=1}^{M} |(\omega_i)_{j,k}|^2,$$

with

$$\hat{\lambda} \equiv \left(\begin{array}{cccc} \hat{\lambda}_{0,0} & \hat{\lambda}_{1,0} & \cdots & \hat{\lambda}_{N-1,N-1} \end{array}\right)^T.$$

Assuming that z is Gaussian, and being careful to properly account for the fact that ω_i is complex, we find that the estimation error variance satisfies

$$E\left((\hat{\lambda}_{i,j}-\lambda_{i,j})(\hat{\lambda}_{k,l}-\lambda_{k,l})\right) = \begin{cases} (\lambda_{i,j}^2)/N_{samp} & \text{if } (i,j)=(k,l), \text{ or} \\ & (i,j)=(N-k,N-l) \\ 0 & \text{otherwise} \end{cases} \tag{2.22}$$

Turning back now to the estimation of r, we see that (2.21) constitutes an unbiased estimate of r. Combining this fact with (2.22), we can bound the diagonal elements of the error covariance (again assuming that z is Gaussian) in the following way:

$$\begin{aligned} E\left((\hat{r}_{m,n}-r_{m,n})^2\right) &= \frac{1}{N^4}\sum_{k,l}\sum_{p,q}\theta_k^m\theta_l^n\theta_p^m\theta_q^n\, E\left((\hat{\lambda}_{k,l}-\lambda_{k,l})(\hat{\lambda}_{p,q}-\lambda_{p,q})\right) \\ &\le \frac{1}{N^4}\sum_{k,l}\sum_{p,q}\left|E\left((\hat{\lambda}_{k,l}-\lambda_{k,l})(\hat{\lambda}_{p,q}-\lambda_{p,q})\right)\right| \\ &= \frac{1}{N^4}\sum_{k,l}\sum_{p,q}E\left((\hat{\lambda}_{k,l}-\lambda_{k,l})(\hat{\lambda}_{p,q}-\lambda_{p,q})\right) \end{aligned}$$

$$\leq \frac{2}{N_{samp}N^4}\sum_{k,l}\lambda_{k,l}^2. \tag{2.23}$$

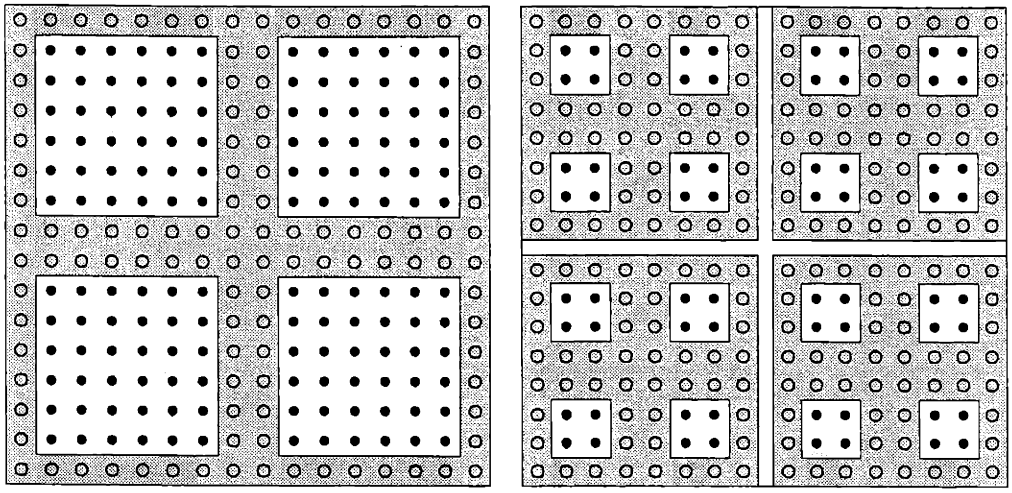
This bound is fairly tight, and is quite useful for determining the number of samples N_{samp} required to increase our confidence in the estimate $\hat{R}_{zz}(\cdot,\cdot)$ to any desired level. The line of reasoning is identical to that used in our foregoing discussion of Monte-Carlo simulation to estimate mean square error of a linear least-squares estimator.

2.3 Multiscale Representations of WS Reciprocal Processes and MRFs

Corresponding to any given zero-mean, WS reciprocal process or MRF, defined on a discrete lattice, is a multiscale process that matches the first and second-order statistics of the given process or field. This fact was proved by construction in [45]. We hasten to add, however, that the existence of these multiscale representations does not completely mitigate the computational complexity problems that we discussed in Chapter 1, in relation to MRFs. The difficulty is that the exact representation of any given WSMRF requires a multiscale model in which the dimension of the states $x(s)$ is quite high; as we saw in Section 2.1.6, a high order places a corresponding high computational burden on signal processing algorithms, thereby limiting the model's utility.

Nevertheless, there is a distinct advantage to the multiscale framework. In particular, the high order, exact representation of a given MRF leads quite naturally to a family of lower-order *approximate* representations. These approximations preserve most of the qualitative and statistical features of the much more complex exact representation. At the same time, they allow us to carry out, in an extremely efficient manner, image processing algorithms that are statistically optimal with respect to the low-order multiscale model and that are nearly optimal with respect to the nearby WSMRF. This claim is substantiated in [43,44], where the authors consider a texture discrimination application. Our main point is that since in most any application, a WSMRF model is itself an idealization, there is a reasonable possibility that aside from the computational speed advantages, our low-order multiscale model will lead to better image restoration, better image segmentation, and so forth.

An exact representation of a WSMRF with the type of tree model described by (2.1) involves a generalization of the mid-point deflection technique for constructing a sample path of 1-D Brownian motion. To construct a Brownian motion sample path over an interval by mid-point deflection, we start by randomly choosing values for the process at the two boundary points of the interval according to the joint probability distribution implied by the Brownian motion model. We then use these two values to compute the expected value of the process at the mid-point, and then add to that a Gaussian random variable with zero mean and variance equal to the variance of the error in this mid-point prediction. The process is then continued by using the original boundary points and newly constructed mid-point to generate values of the Brownian motion at the one-fourth and three-fourths points of the interval. Since Brownian motion is a Markov process, its value at the one-fourth point, conditioned on the values at the initial point and mid-point is *independent* of the process values at the three-fourths and end-points of the interval; a similar conditional independence property holds for the three-fourths point. Thus, we can iteratively generate values at an increasingly dense set of dyadic points in the interval; at each level, we can generate values at



(a) (b)

Figure 2-3: In these figures, we depict the information represented in the state vectors at the top two levels of an exact multiscale representation of a WSMRF defined on a 16 × 16 lattice. At the left is a depiction of the state at the root node, which contains the values of the process at the shaded points. At the right is a depiction of the four states at the second level; for example, the state in the north-west corner contains the values of the process at the shaded points in the north-west 8 × 8 quadrant.

the mid-points of all neighboring pairs of points, in parallel and independently of previously generated points.

A 2-D generalization of the mid-point deflection technique is the key to representing a WSMRF with a multiscale model. In 1-D, we iteratively partitioned the domain of the process by mid-points, and in 2-D, we do this partitioning by *mid-lines*. We define the multiscale model state $x(s)$ to be the set of values of a WSMRF along an appropriately chosen boundary so that the domain of the process is partitioned into smaller, conditionally uncorrelated subdomains. Each of these subdomains is in turn partitioned into even finer subdomains, with each state at the next finer scale corresponding to the values along the boundary of one of these finer subdomains. In this way, we can iteratively generate values of the process along an increasingly dense set of boundaries until the full field has been generated.

Example

By means of a concrete example, we now illustrate the simplicity of the underlying ideas in the foregoing recipe for construction of multiscale WSMRF models. Consider a 2-D WSMRF $z(i,j)$ defined on a 16 × 16 lattice. An exact multiscale model for the GMRF is defined on a quadtree (i.e., $q = 4$). The state vector $x(0)$ at the root node is defined to contain the values of the WSMRF in the shaded boundary and mid-line points shown in Figure 2-3a. The covariance $P(0)$ for this root node is characterized by the joint pdf of the associated WSMRF values.

At the second level, the states consist of points carried down from the root node as well as new mid-line points within each quadrant; the information content of the resulting state vectors is displayed in Figure 2-3b. As we have already remarked, the Markov property ensures that the new mid-lines are conditionally independent given the state of the process at the root node; consequently, these new mid-lines can be constructed independently and

in parallel, using a white-noise driven model of the form (2.1). From the detail provided thus far, the general form of the scale-to-scale recursion should now be clear.

Approximate Multiscale Representations of WSMRFs

Maintaining complete knowledge of a WSMRF process on boundaries of 2-D regions leads to multiscale models in which the state dimension is scale-dependent. For large domains, this dimension can become prohibitively large and so we are led to consider lower order *approximate* representations [43, 45].

At coarse scales, it seems reasonable to retain only coarse approximations to the boundary values. As developed in detail in [45], a logical way to do this is to treat each boundary as a vector of boundary values, and then to do a change of basis, keeping only some of the coefficients in the new basis. We note that by fixing the number of retained coefficient for all levels, we make the state dimension independent of scale. However, since the boundaries are smaller at finer scales, the fixed-dimension "coarse" boundary approximation is in actuality becoming increasingly fine as we move to finer scales.

A natural basis choice, given the multiscale nature of our models is the wavelet basis. The work in [43, 45] demonstrates experimentally the success of this idea in the context of texture representation. In particular, the authors showed that a low-order approximation can yield texture sample paths that are visually indistinguishable from the sample paths of the high order, exact WSMRF. In Chapter 3, we will develop a considerable generalization of these results.

2.4 Canonical Correlation Theory

In this section, we introduce some convenient analytical tools for displaying, in an unambiguous way, the correlation structure between two random vectors having some joint distribution. Historically this material was treated as a somewhat arcane branch of multivariate statistics, known as *canonical correlation theory* [6, 33, 49]. With a more modern perspective [29] has come the realization that this theory is little more than a special application of the *singular value decomposition* (SVD). In a rough sense, the SVD is to these methods what an eigendecomposition is to standard methods for displaying the structure of covariance matrices. When treated with the SVD, canonical correlation theory is both elegant and simple. Moreover, it has practical use; in Chapters 3 and 5, we will adapt it to our needs, in an apparently novel way, to produce a powerful computational engine for our realization algorithms.

2.4.1 Setup and Main Proposition

Let η be a zero-mean random vector, having $(n_1 + n_2)$ components and covariance matrix P_η. We partition η into two sub-vectors, having respectively n_1 and n_2 components,

$$\eta = \begin{pmatrix} \eta_1^T & \eta_2^T \end{pmatrix}^T. \tag{2.24}$$

and we similarly partition the covariance matrix,

$$P_\eta = \begin{pmatrix} P_{\eta_1} & P_{\eta_1\eta_2} \\ P_{\eta_1\eta_2}^T & P_{\eta_2} \end{pmatrix}. \tag{2.25}$$

The following well-known Proposition asserts the existence of a pair of transformation matrices T_1 and T_2 that can be used to clearly exhibit the inter-correlations between η_1 and η_2.

To prepare for the statement of the Proposition, we denote the rank of P_{η_1} by m_1, the rank of P_{η_2} by m_2 and the rank of $P_{\eta_1\eta_2}$ by m_{12}; all of these matrices are allowed to be rank deficient.

Proposition 2 *There exist matrices T_1 and T_2, of dimension $m_1 \times n_1$ and $m_2 \times n_2$, respectively, such that*

$$\begin{pmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{pmatrix} \begin{pmatrix} P_{\eta_1} & P_{\eta_1\eta_2} \\ P_{\eta_1\eta_2}^T & P_{\eta_2} \end{pmatrix} \begin{pmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{pmatrix}^T = \begin{pmatrix} I_{m_1} & D \\ D^T & I_{m_2} \end{pmatrix}, \tag{2.26}$$

and

$$\begin{pmatrix} T_1^+ & \mathbf{0} \\ \mathbf{0} & T_2^+ \end{pmatrix} \begin{pmatrix} I_{m_1} & D \\ D^T & I_{m_2} \end{pmatrix} \begin{pmatrix} T_1^+ & \mathbf{0} \\ \mathbf{0} & T_2^+ \end{pmatrix}^T = \begin{pmatrix} P_{\eta_1} & P_{\eta_1\eta_2} \\ P_{\eta_1\eta_2}^T & P_{\eta_2} \end{pmatrix}. \tag{2.27}$$

In these equations, I_{m_i} is an identity matrix of dimension $m_i \times m_i$ (for $i = 1, 2$). The matrix D has dimension $m_1 \times m_2$ and is given by

$$D = \begin{pmatrix} \hat{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{2.28}$$

where $\hat{D}$ is a positive definite diagonal matrix given by

$$\hat{D} = diag(d_1, d_2, ..., d_{m_{12}}), \quad 1 \geq d_1 \geq d_2 \geq ... \geq d_{m_{12}} > 0. \tag{2.29}$$

Finally, T_i^+ is the Moore-Penrose pseudoinverse of T_i, and is given by

$$T_i^+ = P_{\eta_i} T_i^T, \qquad (i = 1, 2). \tag{2.30}$$

We refer to the triple of matrices (T_1, T_2, D) as the *canonical correlation matrices* associated with (η_1, η_2). Results very similar to Proposition 2 can be found in several places, including [6, 49, 50] and [19]. Our constructive proof in Appendix A.1 is patterned most closely after this last source. We include this proof in order to highlight the computational issues, and also because our formulation allows in a natural way for singular covariance matrices, unlike the references just cited. This allowance is not vacuous posturing; we will need this generality when we apply Proposition 2 in Chapters 3 and 5.

A useful, alternative interpretation of Proposition 2 is that there exist matrices T_1 and T_2 such that the random vectors μ_1 and μ_2 defined by

$$\mu_i = T_i \eta_i, \qquad (i = 1, 2)$$

have covariance

$$E\left[\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \begin{pmatrix} \mu_1^T & \mu_2^T \end{pmatrix}\right] = \begin{pmatrix} I_{m_1} & D \\ D^T & I_{m_2} \end{pmatrix}, \tag{2.31}$$

where the form of the diagonal matrix D is given by (2.28). Furthermore, the transformation from (η_1, η_2) to (μ_1, μ_2) is invertible

$$\eta_i = T_i^+ \mu_i, \qquad (i = 1, 2), \tag{2.32}$$

in a mean-square sense,

$$E[(\eta_i - T_i^+ \mu_i)(\eta_i - T_i^+ \mu_i)^T] = \mathbf{0}, \qquad (i = 1, 2). \tag{2.33}$$

We can exploit the relationship between (η_1, η_2) and (μ_1, μ_2) to yield a decomposition of η_1 and η_2 that isolates their correlated component. Because of the utility of this decomposition, we state it as a corollary.

Corollary 1 *There exists a decomposition of the form*

$$\begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix} n + \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix}, \tag{2.34}$$

where n, ν_1, and ν_2 are zero-mean, uncorrelated random vectors The dimension of n, which is the sole component shared by η_1 and η_2, is equal to the rank of $P_{\eta_1 \eta_2}$. The interpolation matrices, H_1 and H_2 are given by

$$\begin{aligned} H_i &= T_i^+ \begin{pmatrix} I_{m_{12}} \\ \mathbf{0} \end{pmatrix} \\ &= P_{\eta_i} \hat{T}_i^T \qquad (i = 1, 2). \end{aligned} \tag{2.35}$$

The second-order statistics of n, ν_1, and ν_2 are given by

$$\begin{aligned} E(nn^T) &= \hat{D}, \\ E(\nu_i \nu_i^T) &= T_i^+ \left(I_{m_i} - \begin{pmatrix} \hat{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \right) (T_i^+)^T \\ &= P_{\eta_i} - H_i \hat{D} H_i^T \qquad (i = 1, 2). \end{aligned}$$

In the sequel, we refer to the foregoing decomposition of η_1 and η_2 as a *canonical correlation decomposition.*

Proof of Corollary: First, we note by inspection that we can decompose μ_1 and μ_2 as follows:

$$\mu_i = \begin{pmatrix} I_{m_{12}} \\ \mathbf{0} \end{pmatrix} n + \tilde{\mu}_i \qquad (i = 1, 2), \tag{2.36}$$

where $n, \tilde{\mu}_1$, and $\tilde{\mu}_2$ are zero-mean, uncorrelated random vectors, with

$$\begin{aligned} E(nn^T) &= \hat{D}, \\ E(\tilde{\mu}_i \tilde{\mu}_i^T) &= I_{m_i} - \begin{pmatrix} \hat{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \qquad (i = 1, 2). \end{aligned} \tag{2.37}$$

Second, we apply the transformation specified by (2.32) to (2.36), thereby yielding the desired result. **QED**.

In our applications, we will typically not require the full transformation matrices T_1 and T_2; in anticipation of developments later in this paper, we introduce truncated versions of these matrices, denoted by $T_{i,k}$ and defined to contain the first k rows of T_i:

$$T_{i,k} \equiv \begin{pmatrix} I_k & \mathbf{0} \end{pmatrix} T_i \qquad (i = 1, 2).$$

As a special case, we define

$$\hat{T}_i \equiv T_{i,m_{12}}.$$

There are two final comments that we want to make about Proposition 2. First, the proof of the proposition will reveal that there is some flexibility in the choice of T_1 and T_2; these matrices are not unique. On the other hand, all possible choices for T_1 and T_2 lead to the same diagonal matrix D. The following Proposition precisely formulates this uniqueness of D.

Proposition 3 *Let W_1 and W_2 be matrices of dimension $m_1 \times n_1$ and $m_2 \times n_2$, respectively, such that*

$$W_i P_{\eta_i} W_i^T = I_{m_i} \qquad (i = 1, 2)$$

Then, for all such W_1 and W_2, the nonzero singular values of $W_1 P_{\eta_1\eta_2} W_2^T$ are given by the diagonal entries of the matrix $\hat{D}$, which is unique.

Appendix A.2 contains a simple proof of this result. In the sequel, we refer to $\hat{D}$ as the matrix of *canonical correlations*. As we next discuss, these correlations can be given a nice geometric interpretation.

2.4.2 Geometric Interpretations of Diagonal Matrix $\hat{D}$

To bring out the geometric content of the matrix $\hat{D}$ (see (2.28)), we begin by interpreting the scalar components of the random vector η_1 as vectors in a vector space. In particular, we define the vector space $\mathcal{E}_1$ to comprise all linear combinations of the scalar components of the random vector η_1:

$$\mathcal{E}_1 = \left\{ c^T \eta_1 \; ; \; c \in \mathcal{R}^{n_1} \right\}.$$

In a completely analogous way, we define $\mathcal{E}_2$ to be the vector space comprising all linear combinations of the scalar components of the random vector η_2.

Our next step is to capture, in a mathematically precise way, the relative orientation between the two vector spaces $\mathcal{E}_1$ and $\mathcal{E}_2$. A convenient way to do this is in terms of the *principal angles* between the two subspaces [29]. After defining these angles, we will see that each diagonal entry in the matrix $\hat{D}$ is equal to the cosine of a principal angle.

We define the first principal angle θ_1 between $\mathcal{E}_1$ and $\mathcal{E}_2$ to be

$$\begin{aligned} \cos\theta_1 &= \max_{U \in \mathcal{E}_1} \max_{V \in \mathcal{E}_2} \{E(UV)\} \\ &= \max_{c \in \mathcal{R}^{n_1}} \max_{d \in \mathcal{R}^{n_2}} \left\{ c^T P_{\eta_1\eta_2} d \right\}, \end{aligned}$$

where the maximizations are subject to the constraints

$$\begin{aligned} E(U^2) &= c^T P_{\eta_1} c = 1, \\ E(V^2) &= d^T P_{\eta_2} d = 1. \end{aligned}$$

Assume that the maximum is attained for

$$\begin{aligned} U = U_1 &= c_1^T \eta_1, \\ V = V_1 &= d_1^T \eta_2. \end{aligned}$$

We can then define θ_2 as the smallest angle between the orthogonal complement of $\mathcal{E}_1$ with respect to U_1 and the orthogonal complement of $\mathcal{E}_2$ with respect to V_1. We continue in this way until one of the spaces $\mathcal{E}_1$, $\mathcal{E}_2$ is reduced to $\{0\}$; this will happen after a number of steps equal to the minimum of m_1 and m_2. Thus, we recursively define θ_k, U_k, V_k, c_k, and d_k by

$$\begin{aligned} \cos\theta_k &= \max_{U \in \mathcal{E}_1} \max_{V \in \mathcal{E}_2} \{E(UV)\} \\ &= \max_{c \in \mathcal{R}^{n_1}} \max_{d \in \mathcal{R}^{n_2}} \left\{ c^T P_{\eta_1 \eta_2} d \right\}, \end{aligned}$$

where the maximizations are subject to the constraints

$$\begin{aligned} E(U^2) &= c^T P_{\eta_1} c = 1, & \\ E(U_j U) &= c_j^T P_{\eta_1} c = 0, & j = 1, 2, ..., k-1, \\ E(V^2) &= d^T P_{\eta_2} d = 1, & \\ E(V_j V) &= d_j^T P_{\eta_2} d = 0, & j = 1, 2, ..., k-1, \end{aligned}$$

and we assume that the maximum is attained for

$$\begin{aligned} U = U_k &= c_k^T \eta_1, \\ V = V_k &= d_k^T \eta_2. \end{aligned}$$

These constrained maximizations may appear quite formidable to solve, and in fact, historically, they were solved via the laborious route of using Lagrange multipliers [6, 33]. On the other hand, with a more modern perspective has come the realization that the SVD can often simplify these kinds of vector space issues [62]. The SVD is exploited in the proof of Proposition 2, which in turn manages all of the difficult work needed to find principal angles. The relation between these angles and D is as follows:

$$\cos\theta_k = \begin{cases} d_k & k = 1, 2, ..., m_{12}, \\ 0 & k = m_{12}+1,\ m_{12}+2, ...,\ \min(m_1, m_2), \end{cases}$$

where $d_1, d_2, ..., d_{m_{12}}$ are defined in (2.29). Although the proof of this result is straightforward, we omit it and instead refer the reader to [50].

2.4.3 Computational Issues

The proof of Proposition 2 reveals that in general, for covariance matrix P_η of dimension $N \times N$, the computational complexity of calculating the canonical correlation matrices $(\hat{T}_1, \hat{T}_2, \hat{D})$ is roughly $\mathcal{O}(N^3)$ floating point operations. The computations involved can be

summarized as follows:

Step 1. Carry out two eigendecompositions: $P_{\eta_i} = S_i \Lambda_i S_i^T \qquad (i = 1, 2)$.
Step 2. Carry out SVD calculation: $\Lambda_1^{-1/2} S_1^T P_{\eta_1 \eta_2} S_2 \Lambda_2^{-1/2} = U_1 D U_2^T$.
Step 3. Set $T_i = U_i^T \Lambda_i^{-1/2} S_i^T$, $(i = 1, 2)$.

For problems of practical interest to us, the dimension N could easily be on the order of a million, in which case the computational complexity $\mathcal{O}(N^3)$ is absolutely prohibitive. Thus, given the importance of the canonical correlation decomposition in our later work, we have considerable motivation for seeking intelligent ways to reduce this computational burden.

If the correlation between η_1 and η_2 has a certain special structure, then we can achieve a substantial reduction in the complexity of the computation of $\left\{\hat{T}_1, \hat{T}_2, \hat{D}\right\}$. In a loose sense, this structure can be described in the following way. We assume that the correlated component of η_1 and η_2 lives in some low-dimensional subspace that is easily defined; we then do all of our computations with low-dimensional random vectors that live in this subspace, and thereby achieve our complexity reduction.

To describe more carefully this assumed structure of the correlation between η_1 and η_2, we introduce the two matrices Θ_1 and Θ_2, which are assumed to capture the correlated component of η_1 and η_2 in the sense that

$$E\left[(\eta_1 - E(\eta_1|\mu_i))(\eta_2 - E(\eta_2|\mu_i))^T\right] = \mathbf{0}, \qquad (i = 1, 2). \tag{2.38}$$

where

$$\mu_i = \Theta_i \eta_i, \qquad (i = 1, 2). \tag{2.39}$$

The following proposition summarizes the key result.

Proposition 4 *Let $(\hat{T}_1, \hat{T}_2, \hat{D})$ be the canonical correlation matrices for (μ_1, μ_2). If (μ_1, μ_2) are related to (η_1, η_2), as in (2.38) and (2.39), then $(\hat{T}_1\Theta_1, \hat{T}_2\Theta_2, \hat{D})$ are the canonical correlation matrices for (η_1, η_2).*

A proof is contained in Appendix A.3. The implications of this result are explored in Chapter 3, where the result is used extensively in our model-building procedure.

Chapter 3

Multiscale Stochastic Realization

In order for the multiscale framework to realize its full potential as a powerful approach for solving statistical signal processing problems, there is a fundamental need for systematic model-building tools. Just as Kalman filtering requires the prior specification of a state-space model, so do our multiscale estimation algorithms require such a prior specification. While we have seen in Section 2.1.2 that the second-order statistics of a multiscale process $x(s)$ can be readily determined from given values for the model parameters $P(0)$, $A(s)$ and $B(s)$, the converse is not necessarily true. In particular, it is generally quite challenging to *devise* values for $P(0)$, $A(s)$ and $B(s)$ so that they yield a given specification of the second-order statistics of $x(s)$. This latter, more challenging problem is the focus of this chapter.

3.1 Introduction

Our approach to multiscale stochastic realization is based on a synthesis of ideas from two distinct sources: (i) Akaike's work in [2] on building state-space models for stationary time series via canonical correlation analysis, and (ii) the work in [45] on building multiscale models for WSMRFs. Let us consider the influence of each of these sources, in turn.

In [2], Akaike considered the problem of building a minimal-dimension state-space model to realize a stationary random process $y(n)$ $(n = \ldots, -1, 0, \ldots)$, given knowledge of the corresponding covariance matrices $R_{yy}(l)$ $(l = 0, 1, \ldots)$,

$$R_{yy}(l) \equiv E\left[y(n+l)y^T(n)\right]. \tag{3.1}$$

The structure of the models he sought have the following familiar state-space form, discussed in Chapter 1:

$$\begin{aligned} z(n+1) &= Az(n) + w(n) \\ y(n) &= Cz(n), \end{aligned} \tag{3.2}$$

where $z(n)$ denotes the state of the process at time n, A is the one-step transition matrix, C is the observation matrix and $w(n)$ is a zero-mean, white Gaussian noise driving term.

Akaike motivated his approach to building these models by highlighting their *Markov* property: conditioned on the present value of $z(n)$, the past values of y (i.e., $y(n-1), y(n-2), \ldots$) are statistically independent of the future values of y (i.e., $y(n+1), y(n+2), \ldots$). In

this sense, the role of the state information in $z(n)$ is to provide an interface between the past and the future. Of course, the dimension of the needed interface will be closely tied to the structure of the covariance matrices $R_{yy}(l)$, and for many applications, one can expect that an exact realization of the specified $R_{yy}(l)$ will require an unduly high state dimension. For instance, in a Kalman filtering application, the computational load per time instant n is proportional to the cube of the dimension of the state $z(n)$, and if the model order grows too large, then the algorithm becomes too slow. Thus, there are two principal issues that must be confronted to deal with this time-series realization problem in a satisfactory manner. First, for exact realizations, a method is needed for finding the minimal dimension and corresponding information content of the state. Second, for reduced-order, approximate realizations, a method is needed for measuring the relative importance of the components of the information interface provided by the state, so that a decision can be made about which components to discard in a reduced-order realization.

In the multiscale context, the realization issues are very similar, only now, the state must act as an information interface among *multiple* subsets of the process, not just two. To see clearly the role of state information in the multiscale context, let us consider a multiscale process indexed on a q-th order tree. As we know from Chapter 2, the state $x(s)$ at any given node of such a tree represents an appropriate, aggregate description of the subset of the finest-scale process that descends from the given node. Furthermore, the given node partitions the remaining nodes into $(q+1)$ disjoint subtrees, one associated with each of the children and parent nodes. This partitioning property leads us to the important point that just as with time-series models, our multiscale models have a Markov property. This property was described in Chapter 2, but its importance compels us to re-state it here: if $x(s)$ is the value of the state at node s, then conditioned on the value of $x(s)$, the values of the states in the corresponding $q+1$ subtrees of nodes extending away from s are uncorrelated. In light of the Markov property, the role of the state in a multiscale process is clear. In particular, the role of the state at any node is to store enough information about the process to decorrelate the corresponding $q+1$ subsets of the process. Moreover, continuing the parallel with the time-series case, complete retention of the needed decorrelating information may lead to state vectors of unacceptably high dimension, and thus we are frequently motivated to turn to reduced-order, approximate realizations.

As Akaike's work showed, there is an elegant way to deal simultaneously and coherently with the issues of both exact and reduced-order modeling. The idea is to bring to bear *canonical correlation analysis*, which was reviewed in Chapter 2. While in its original form, this tool is helpful only for the static problem of unambiguously displaying the correlation structure between two random vectors, Akaike adapted this tool to a dynamical context, and thereby devised his state-space models.

By another appropriate adaptation, we too exploit canonical correlation analysis to build state-space models, but now for our multiscale processes. In making the adaptation to this new dynamical context, we are guided by the WSMRF work in [45], which was the first to highlight and exploit the decorrelating role of state information in building multiscale models. For our purposes, the most important characteristic of this work is its systematic decomposition of the modeling problem into a collection of smaller, independent sub-problems, each myopically focused on designing the information content of a single state vector to fulfill that vector's designated interfacing role. It turns out that once this information content has been determined, the rest of the model parameters follow readily. By formalizing and generalizing this decomposition of the modeling problem, we are led naturally to a procedure that allows us to build models for a wider class of random processes

and fields than just Markov ones.

To carry out this effort, we begin by restricting the value of the state vector associated with each node to be a linear function of the finest-scale process or field that descends from the given node. In this way, just as in [45], we reduce the realization problem to one of determining the linear function (i.e., the matrix) that relates the state vector and the finest-scale process or field.

After reducing the modeling problem to a collection of independent sub-problems, we next make precise the notion of a state vector *approximately* fulfilling its decorrelating role. Specifically, we introduce a measure of decorrelation that is closely related to the principal angle concepts of Section 2.4.2. We then demonstrate that with respect to this metric, canonical correlation analysis can in principle be used to optimize the content of the state vectors; that is, canonical correlation analysis can be used to solve for the matrices parameterizing the state vectors. In practice, an exact analysis is prohibitive, because of the high dimension of the relevant covariance matrices, and so approximations are required to manage the SVD calculations. We devise a particular approximation scheme that is motivated directly by the simplifications possible when the random process or field to be modeled is a WS Markov. Thus, our canonical correlation calculations are exact for WS Markov processes and fields, and are approximate otherwise. Our experimental results will demonstrate that these approximations are quite effective. Ultimately we obtain a flexible, general model-building algorithm that it is capable of generating accurate and useful models for both Markov and non-Markov random processes and fields.

This chapter is organized in the following way. We begin by providing a brief overview of Akaike's approach to the time-series realization problem. We then set up the multiscale modeling problem, and reduce it formally to one of determining the information content of each state vector. Next we make precise the notion of approximate realizations, and we develop our solution to the modeling problem, where this solution takes the form of a readily implementable algorithm having canonical correlation analysis as its computational engine. We illustrate the application of the algorithm, by building multiscale models for both processes and fields. Finally, we conclude with a rather lengthy discussion of an interesting and non-trivial difference, uncovered by our analysis, between time-series stochastic processes and multiscale stochastic processes. This difference is in regard to minimal-dimension models and their expressibility in terms of so called *internal* realizations. As will be made precise in our detailed discussion, the class of internal realizations is sufficiently rich in the time-series context to always include a minimal model, while the same is not true in the multiscale context, where sometimes a so-called *external* realization is required to build a minimal model. This fact will lead us to return in the last part of this chapter to the modeling problem, to develop further tools, explicitly tailored to build external realizations.

3.2 Akaike's Approach to Stochastic Realization of Stationary Time Series

As described in the previous section, the objective in the time-series realization problem is to build a minimal-dimension state-space model of the form (3.2), driven by white noise, so that the output process $y(n)$ has some given correlation $R_{yy}(i)$. Let us briefly consider Akaike's approach to this problem. The insights we obtain we will be quite useful for addressing the multiscale counterpart of this realization problem.

One of Akaike's achievements was to characterize the minimum system dimension (i.e.,

the minimum dimension of the state $z(n)$) that can be used to realize $y(n)$. Furthermore, he devised a particular coordinate choice that leads to an unambiguous arrangement of the components of the state-vector in descending order of importance to the past/future interface, thus facilitating any decision about which components to discard in a reduced-order realization. To elaborate on these results, we let $\eta_{past}(n)$ be an infinite dimensional random vector containing the values of the process $y(n)$ for all times less than or equal to n, and let $\eta_{future}(n)$ be an infinite dimensional random vector containing the values of the process $y(n)$ for all times greater than or equal to n:

$$\eta_{past}(n) = \begin{pmatrix} \vdots \\ y(n-1) \\ y(n) \end{pmatrix}, \qquad \eta_{future}(n) = \begin{pmatrix} y(n) \\ y(n+1) \\ \vdots \end{pmatrix}. \tag{3.3}$$

In terms of $\eta_{past}(n)$ and $\eta_{future}(n)$, an exact realization of the specified statistics for $y(n)$ requires that the state $z(n)$ be such that

$$\begin{aligned} &\tilde{\eta}_{past}(n) \text{ and } \tilde{\eta}_{future}(n) \text{ are uncorrelated, where} \\ &\tilde{\eta}_{pos}(n) \equiv \eta_{pos}(n) - E(\eta_{pos}(n) \mid z(n)), \quad pos = past,\ future \end{aligned}$$

Letting $(\hat{T}_1,\ \hat{T}_2\ ,\ \hat{D})$ be the canonical correlation matrices[1] associated with $(\eta_{past}(n),\ \eta_{future}(n))$, Akaike demonstrated that the minimal-dimension interface $z(n)$ is equal to the number of non-zero elements of $\hat{D}$. Two possible choices for the state vector $z(n)$ are

$$z(n) = \hat{T}_2\eta_{future}(n), \quad \text{and} \quad z(n) = E(\hat{T}_2\eta_{future}(n) \mid \hat{T}_1\eta_{past}(n)), \tag{3.4}$$

where the components of both are indeed arranged in descending order of importance to the past/future interface; this prioritized ordering can be shown to hold with respect to both an information-theoretic criterion, as in [27], and with respect to the generalized correlation coefficient criterion of Section 3.3.3.

In establishing (3.4), Akaike managed the infinite dimension of $\eta_{past}(n)$ and $\eta_{future}(n)$ by effectively using a result akin to our Proposition 4 of Chapter 2, thereby creating finite-dimensional vectors $\mu_{past}(n)$ and $\mu_{future}(n)$ that entirely capture the correlated component of $\eta_{past}(n)$ and $\eta_{future}(n)$. In general, $\mu_{past}(n)$ can be defined as follows:

$$\begin{aligned} \mu_{past}(n) = (\ &E^T(y(n) \mid \eta_{past}(n)) \quad E^T(y(n+1) \mid \eta_{past}(n)) \quad \cdots \\ &E^T(y(n+r-1) \mid \eta_{past}(n)))^T \end{aligned}$$

and similarly, $\mu_{future}(n)$ can be defined as

$$\begin{aligned} \mu_{future}(n) = (\ &E^T(y(n) \mid \eta_{future}(n)) \quad E^T(y(n-1) \mid \eta_{future}(n)) \quad \cdots \\ &E^T(y(n-r+1) \mid \eta_{future}(n)))^T \end{aligned}$$

where r is any finite upper-bound on the system dimension. In the language of [3], linear combinations of the random vectors $\mu_{past}(n)$ and $\mu_{future}(n)$ are said to span the *predictor* space of the past and future, respectively. A simplification is possible if $y(n)$ is an r-th

[1] Thanks to the stationarity of $y(n)$, the matrices $\hat{T}_1$ $\hat{T}_2$ and $\hat{D}$ are here independent of time.

order WS Markov process; we can then define $\mu_{past}(n)$ and $\mu_{future}(n)$ as

$$\mu_{past}(n) = \begin{pmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-r+1) \end{pmatrix}, \qquad \mu_{future}(n) = \begin{pmatrix} y(n) \\ y(n+1) \\ \vdots \\ y(n+r-1) \end{pmatrix},$$

Consistent with Proposition 4, then, the non-zero canonical correlation coefficients between $\eta_{past}(n)$ and $\eta_{future}(n)$ are given by the non-zero canonical correlation coefficients between $\mu_{past}(n)$ and $\mu_{future}(n)$.

It turns out that once the information content of $z(n)$ has been defined, as for example by (3.4), then the parameters of the state-space model (3.1) follow readily [2]. Thus, attention can be focused on designing the content of $z(n)$. This reduction of the problem is noteworthy, because in the multiscale context, we will focus attention in a similar fashion, and in fact will use a natural extension of these canonical correlation tools to determine the information content of the state vectors.

3.3 Formulation of Realization Problem

Let us suppose we have a q-th order tree, consisting of $M+1$ scales. For random fields (representing imagery, for instance), a quadtree as in Figure 1-1 would typically used, while for random processes, a dyadic tree as in Figure 2-1 would typically be used. We emphasize, though, that we will be treating all tree orders in a uniform way, allowing for all orders $q \geq 2$.

Informally, our objective is to build a multiscale model, indexed on the given tree, such that the covariance of the resulting finest-scale process comes as close as possible to matching some prespecified covariance. We must devise values for the following model parameters:

1. the covariance $P(0)$ of the state at the root node,

2. the interpolation matrices $A(s)$, and

3. the noise-shaping matrices $B(s)$.

As detailed in Sections 2.1.1, these parameters provide a complete specification of the process.

To ensure that the constructed multiscale model achieves a balance between the possibly conflicting requirements of low dimension and high fidelity, we impose further structure on the realization problem. There are two natural ways to do this structuring, and we consider both in parallel throughout this chapter. In one formulation, we fix the quality of the match between prespecified and actual realized finest-scale covariances, and subject to this constraint, we seek a multiscale model having state vectors of the minimum possible dimension. In the other formulation, we constrain the model dimension (i.e., the dimension of the process state vectors $x(s)$), and subject to this alternative constraint, we seek a multiscale model having the best match between prespecified and actual, realized finest-scale covariances. For now, we treat only informally the notion of this covariance matching, deferring precise statements until Sections 3.3.3.

3.3.1 Notation

To maintain a clear distinction between the specified, desired statistics and the actual, realized ones, we use different notation for each. We denote the prespecified finest-scale covariance by P_{χ_0}, where χ_0 is a random vector having this covariance. Correspondingly, as in Section 2.1.4, we denote the actual, realized covariance by P_{ξ_0}, where ξ_0 is a random vector containing the finest-scale state information in this realized model. Also as in Section 2.1.4, we let ξ_s and ξ_{s^c} be the particular sub-vectors of ξ_0 that contain, respectively, the finest-scale state information that does and does not descend from s. In an analogous manner, we define the random vectors χ_s and χ_{s^c} as sub-vectors of χ_0. We denote the covariance of ξ_s by P_{ξ_s} and the cross-covariance between ξ_s and ξ_σ by $P_{\xi_s\xi_\sigma}$. Finally, we denote the covariance of χ_s by P_{χ_s} and the cross-covariance between χ_s and χ_σ by $P_{\chi_s\chi_\sigma}$.

3.3.2 Parameterizing content of $x(s)$ by W_s

As we work our way toward a more careful statement of the modeling problem, we wish to shift the emphasis away from determining the parameters $P(0)$, $A(s)$ and $B(s)$ to focus instead on designing the information content of the state vectors $x(s)$. Such a shift was central to both Akaike's approach to the time-series realization problem and also the approach in [45] to building multiscale representations of WSMRFs; by suitably generalizing these results, we will find that such a shift is possible in our context as well. The primary benefit of proceeding this way is our subsequent ability to reduce the realization problem to a collection of independent sub-problems, each myopically focused on determining the information content of a single state vector $x(s)$ to fulfill its decorrelating role.

Multiscale representations of WSMRFs

To effect our desired shift in focus, let us begin by examining in greater detail the construction in [45] of multiscale models for WSMRFs. As overviewed in Section 2.3, the key in that context is to define the multiscale process state $x(s)$ to contain the set of values of the MRF along an appropriately chosen boundary, so that the MRF region is partitioned into smaller, conditionally uncorrelated sub-regions. To be specific, let us consider a 2-D WSMRF defined on a $2^M \times 2^M$ lattice, for which we correspondingly consider a multiscale representation, indexed on a quadtree. The root-node state $x(0)$ can fulfill its decorrelating role if it contains the values of the MRF around the outer boundary of the lattice and also along the vertical and horizontal mid-lines. For instance, on a 16×16 lattice, the root state $x(0)$ should contain the values of the MRF at the shaded boundary and mid-line points shown in Figure 2-3a. Letting χ_0 denote the MRF, this strategy for defining $x(0)$ leads to the following decorrelating property:

$$\begin{aligned} &\tilde{\chi}_{0\alpha_i|0} \perp \tilde{\chi}_{(0\alpha_i)^c|0}, \qquad i = 1,2,3,4, \\ &\tilde{\chi}_{\sigma|0} \equiv \chi_\sigma - E(\chi_\sigma \mid x(0)), \quad \sigma = 0\alpha_1, \ldots, 0\alpha_4, \end{aligned}$$

which we will soon see in a more general guise in our more general modeling strategy. Furthermore, this choice for $x(0)$ implies that $P(0)$ must be given by

$$\begin{aligned} P(0) &= E\left[x(0)x^T(0)\right] \\ &= W_0 P_{\chi_0} W_0^T, \end{aligned} \tag{3.5}$$

where W_0 is a selection matrix, such that for $x(0) = W_0\chi_0$, $x(0)$ contains the appropriate boundary information.

Now, turning to the construction of the next scale of the multiscale representation of the WSMRF, the components of the four state vectors at this scale should contain appropriate boundary information to partition the MRF region into even finer sub-regions. This idea is illustrated in Figure 2-3b for our example involving the 16×16 lattice; in this case, the shaded grid points comprise the components of the four state vectors. Here, the important point is that the values of these four state vectors can be generated, *independently and in parallel* using a white-noise driven model of the form (2.1). In particular, for $i = 1$, 2, 3 and 4, we can decompose $x(0\alpha_i)$ as

$$x(0\alpha_i) \;=\; E\left[x(0\alpha_i) \mid x(0)\right] + \tilde{x}(0\alpha_i), \tag{3.6}$$

where, thanks to (3.5), $\tilde{x}(0\alpha_1), \ldots, \tilde{x}(0\alpha_4)$ are uncorrelated; we can then recast (3.6) as

$$x(0\alpha_i) \;=\; A(0\alpha_i)x(0) + B(0\alpha_i)w(0\alpha_i),$$

exactly as in (2.1), with $A(0\alpha_i)\ x(0)$ representing the prediction term $E\left[x(0\alpha_i) \mid x(0)\right]$, and $B(0\alpha_i)\ w(0\alpha_i)$ representing the detail term $\tilde{x}(0\alpha_i)$. We emphasize that values for $A(s)$ and $B(s)$ can here be calculated in almost a trivial fashion, once the information content of $x(s)$ and $x(s\bar{\gamma})$ has been devised.

We can iterate the construction just described, by defining states at successive levels to be the values of the WSMRF at boundary and mid-line points of successively smaller subregions. In particular, we can define the state $x(s)$ to have the form

$$x(s) \;=\; W_s\chi_s, \tag{3.7}$$

where W_s is a selection matrix chosen to fulfill the following decorrelating condition:

$$\begin{array}{l} \tilde{\chi}_{s\alpha_i|s} \perp \tilde{\chi}_{(s\alpha_i)^c|s}, \qquad i = 1, 2, \ldots, q \\ \tilde{\chi}_{\sigma|s} \;\equiv\; \chi_\sigma - E(\chi_\sigma \mid W_s\chi_s), \quad \sigma \;=\; s\alpha_1, s\alpha_2, \ldots, s\alpha_q. \end{array} \tag{3.8}$$

Each of these state vectors can be related to the state at its parent node by a white-noise driven model of the form (2.1). Specifically, we have that

$$\begin{aligned} E[x(s) \mid x(s\bar{\gamma})] \;&=\; E(W_s\xi_s \mid W_{s\bar{\gamma}}\xi_{s\bar{\gamma}}) \\ &=\; W_sP_{\xi_s\xi_{s\bar{\gamma}}}W_{s\bar{\gamma}}^T \left\{W_{s\bar{\gamma}}P_{\xi_{s\bar{\gamma}}}W_{s\bar{\gamma}}^T\right\}^{-1} x(s\bar{\gamma}), \end{aligned}$$

and

$$\begin{aligned} E[\tilde{x}(s)\tilde{x}^T(s)] \;&=\; E[x(s)x^T(s)] - E[x(s)x^T(s\bar{\gamma})]E^{-1}[x(s\bar{\gamma})x^T(s\bar{\gamma})]E[x(s\bar{\gamma})x^T(s)] \\ &=\; W_sP_{\xi_s}W_s^T \;- \\ &\qquad W_sP_{\xi_s\xi_{s\bar{\gamma}}}W_{s\bar{\gamma}}^T \left\{W_{s\bar{\gamma}}P_{\xi_{s\bar{\gamma}}}W_{s\bar{\gamma}}^T\right\}^{-1} W_{s\bar{\gamma}}P_{\xi_{s\bar{\gamma}}\xi_s}W_s^T. \end{aligned}$$

Hence, to maintain consistency with (2.1), we set

$$A(s) = W_sP_{\xi_s\xi_{s\bar{\gamma}}}W_{s\bar{\gamma}}^T \left\{W_{s\bar{\gamma}}P_{\xi_{s\bar{\gamma}}}W_{s\bar{\gamma}}^T\right\}^{-1}, \tag{3.9}$$

and

$$B(s)B^T(s) = W_s P_{\xi_s} W_s^T - A(s) W_{s\bar{\gamma}} P_{\xi_{s\bar{\gamma}}\xi_s} W_s^T. \tag{3.10}$$

Together, (3.5), (3.9) and (3.10) provide all the model parameters needed to specify the multiscale model, and they can all be determined readily from the arguably more fundamental W_s parameters.

Generalizing the WSMRF construction

The applicability of the foregoing construction can be broadened considerably, by making a single modification. In particular, there is no reason to limit attention to selection matrices for W_s; while such a choice was certainly appropriate and natural for WSMRFs, it is clear that a linear parameterization of the form (3.7) can accommodate a far richer class of state vectors than merely decimated versions of the finest-scale process. In fact, a careful study of the preceding construction will reveal that *any* given covariance matrix P_{χ_0} can be realized as the finest scale of a multiscale process, by carrying out a two-stage procedure: (i) determination of a matrix W_s that fulfills (3.8), for each node[2] s, with all the matrices at the finest scale being identity matrices, and (ii) calculation of $P(0)$, $A(s)$ and $B(s)$ via (3.5), (3.9) and (3.10), respectively. This procedure will yield a multiscale model for which $P_{\xi_0} = P_{\chi_0}$.

The difficulty with the procedure just outlined is that it is only applicable to building exact realizations, which almost always suffer from an impractically high state dimension. We saw evidence of this fact with exact multiscale representations of WSMRFs, where the required state dimension at, say, the root node was on the order of the number of pixels along the perimeter of the MRF region; for many non-Markov random fields, we expect that this required dimension for exact realizations will be even higher. Thus, the more pressing, practical challenge is to formalize and address the problem of reduced-order modeling.

Fortunately, the WSMRF construction is of considerable utility for obtaining insights into the reduced-order modeling problem as well. The key is to relax the rather stringent condition (3.8) on the W_s matrices, and allow for W_s matrices that only *approximately* fulfill (3.8). We can then apply these matrices in (3.5), (3.9) and (3.10) to obtain parameter values $P(0)$, $A(s)$, and $B(s)$ for a reduced-order model, in which, hopefully, $P_{\xi_0} \approx P_{\chi_0}$. Indeed, this is the approach we pursue.

Regardless of how we actually determine the W_s matrices for a reduced-order model, we can obtain some immediate insight into the nature of the resulting approximation $P_{\xi_0} \approx P_{\chi_0}$. In particular, for every node s, the covariance $P_{x(s)}$ of the state vector $x(s)$ will satisfy

$$P_{x(s)} = W_s P_{\chi_s} W_s^T. \tag{3.11}$$

This fact can be established by induction. Specifically, the validity of (3.11) holds by construction at the root node (see (3.5)); if we assume that for any s, (3.11) is true for node $s\bar{\gamma}$, then by substituting the values for $A(s)$ and $B(s)$ into the Lyapunov equation (2.2), we find that (3.11) must also be true at node s, thus completing the inductive argument. As a consequence of (3.11), the diagonal blocks of P_{ξ_0} can trivially be guaranteed to match the diagonal blocks of P_{χ_0}; we simply let each of the finest-scale W_s matrices be an identity

[2]The choice $W_s = I$ is universally valid, though of virtually no practical value, owing to the high dimension for $x(s)$ to which it leads.

matrix.

Given that the diagonal components of the covariance P_{χ_0} can be matched exactly in P_ξ, the next question is to what extent the off-diagonal correlation can be matched. At this point, the specific details of the W_s matrices become important; in fact, most of the rest of this chapter is devoted to determining values for these matrices. In Sections 3.3.3 and 3.3.4 we make precise the notion of approximate fulfillment of (3.8). Then, in Sections 3.4 and 3.5, we develop in detail algorithms for finding suitable W_s matrices. Finally, we examine further the nature of the approximate equality $P_{\xi_0} \approx P_{\chi_0}$ through our numerical experiments in Section 3.6.

3.3.3 The Generalized Correlation Coefficient

To measure the degree to which a given W_s matrix approximately fulfills (3.8), we introduce a *generalized correlation coefficient*. To define this coefficient, we start with the more standard correlation coefficient $\rho(\eta_1, \eta_2)$ for a pair of scalar valued random variables η_1 and η_2:

$$\rho(\eta_1, \eta_2) \equiv \begin{cases} \frac{E[(\eta_1 - E(\eta_1))(\eta_2 - E(\eta_2))]}{\sqrt{\text{var}(\eta_1)\,\text{var}(\eta_2)}} & \text{if } \text{var}(\eta_i) > 0, \text{ for } i = 1, 2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

Here, $\text{var}(\eta_i)$ denotes the variance of η_i. We then define our generalization of $\rho(\cdot, \cdot)$ in two steps. First, for vector-valued random variables η_1 and η_2, we define their generalized correlation coefficient $\bar{\rho}(\eta_1, \eta_2)$ by

$$\bar{\rho}(\eta_1, \eta_2) \equiv \max_{f_1, f_2} \left\{ \rho(f_1^T \eta_1, f_2^T \eta_2) \right\}$$

where the dummy argument f_i (for $i = 1, 2$) is a column vector having the same dimension as η_i. As an immediate consequence of our discussion in Section 2.4.2, we note that this definition of $\bar{\rho}(\cdot, \cdot)$ implies that

$$\bar{\rho}(\eta_1, \eta_2) = d_1$$

where d_1 is the first canonical correlation coefficient between η_1 and η_2. To extend the definition of $\bar{\rho}(\cdot, \cdot)$ to a collection of random vectors $\eta_1, \eta_2, \ldots, \eta_k$, we proceed in the following way:

$$\bar{\rho}\,(\eta_1, \eta_2, \ldots, \eta_k) \equiv \max_{i \neq j} \bar{\rho}(\eta_i, \eta_j).$$

Each of these correlation coefficients has a conditioned version. In particular, the correlation coefficient between scalar-valued random variables η_1 and η_2, conditioned on (possibly vector-valued) random variable z, is denoted by $\rho(\eta_1, \eta_2 \mid z)$, and is defined as

$$\rho(\eta_1, \eta_2 \mid z) \equiv \begin{cases} \frac{E[(\eta_1 - E(\eta_1|z))(\eta_2 - E(\eta_2|z)) \mid z]}{\sqrt{\text{var}(\eta_1|z)\,\text{var}(\eta_2|z)}} & \text{if } \text{var}(\eta_i|z) > 0, \text{ for } i = 1, 2, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\mathrm{var}(\eta_i|z) \equiv E\left[(\eta_i - E(\eta_i|z))^2 \mid z\right] \quad (i=1,2),$$

and where we recall that $E(x|y)$ denotes the linear least-squares estimate of x given y. The quantities $\bar{\rho}(\eta_1,\eta_2 \mid z)$ and $\bar{\rho}(\eta_1,\eta_2,\ldots,\eta_k \mid z)$ are defined in an analogous manner.

When the conditioning information z in $\bar{\rho}(\eta_1,\eta_2 \mid z)$ is a linear function of η_1 and η_2,

$$z = W\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix},$$

then we can express $\bar{\rho}(\eta_1,\eta_2 \mid z)$ in a particularly convenient form, as we now record for later use. Letting P_η denote the covariance of $\begin{pmatrix}\eta_1^T & \eta_2^T\end{pmatrix}^T$, and defining $\tilde{P}_\eta$ as

$$\tilde{P}_\eta = P_\eta - P_\eta W^T(WP_\eta W^T)^{-1}WP_\eta \tag{3.13}$$

$$\equiv \begin{pmatrix}\tilde{P}_{\eta_1} & \tilde{P}_{\eta_1\eta_2}\\ \tilde{P}_{\eta_1\eta_2}^T & \tilde{P}_{\eta_2}\end{pmatrix}, \tag{3.14}$$

we have that

$$\bar{\rho}\left(\eta_1,\eta_2 \mid W\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix}\right) = \max_{f_1\in F_1,\, f_2\in F_2}\left\{f_1^T\tilde{P}_{\eta_1\eta_2}f_2\right\}, \tag{3.15}$$

where

$$F_i \equiv \left\{f\in\mathcal{R}^{n_i};\ f^T\tilde{P}_{\eta_i}f=1\right\}. \tag{3.16}$$

For future reference, we note two useful identities regarding the generalized correlation coefficient. For one, the unconditioned and conditioned correlation coefficients are related in the following fashion:

$$\bar{\rho}(\eta_1,\eta_2 \mid z_1) = \bar{\rho}(\tilde{\eta}_1,\tilde{\eta}_2), \tag{3.17}$$

where

$$\tilde{\eta}_i = \eta_i - E(\eta_i|z_1),\ (i=1,2). \tag{3.18}$$

This fact can be verified by direct application of the definitions.

Our second (and much more important) result demonstrates that $\bar{\rho}(\cdot,\cdot \mid \cdot)$ is a non-increasing function as the amount of conditioning information increases.

Proposition 5 *For $i=1,2$ and for all matrices W_i,*

$$\bar{\rho}(\eta_1,\eta_2 \mid W_i\eta_i) \leq \bar{\rho}(\eta_1,\eta_2).$$

A proof of Proposition 5 is contained in Appendix B. We emphasize that if the conditioning information is not a function of either η_1 or η_2 alone, then the function $\bar{\rho}(\cdot,\cdot \mid \cdot)$ may become

an increasing one. For instance, if

$$E\left[\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix}\begin{pmatrix}\eta_1^T & \eta_2^T\end{pmatrix}\right] = \begin{pmatrix}1 & 0.5\\ 0.5 & 1\end{pmatrix},$$

then

$$\bar{\rho}(\eta_1, \eta_2) = 0.5,$$

but

$$\bar{\rho}(\eta_1, \eta_2 \mid \eta_1 + \eta_2) = 1.$$

We can, however, strengthen Proposition 5 a bit, by relaxing our restriction that all of the conditioning information be a linear function of either η_1 or η_2; in lieu of this restriction, we restrict each individual scalar component of this conditioning information to be a function of either η_1 or η_2. We state this result as a corollary:

Corollary 2 *For any pair* $(i_1, i_2) \in \{\{1,2\} \times \{1,2\}\}$, *and for all matrices* W_1, W_2,

$$\bar{\rho}(\eta_1, \eta_2 \mid W_1\eta_{i_1}, W_2\eta_{i_2}) \leq \bar{\rho}(\eta_1, \eta_2 \mid W_1\eta_{i_1})$$

By combining (3.17) with Proposition 5, the validity of the corollary becomes immediate.

3.3.4 Precise casting of condition on W_s matrices

The generalized correlation coefficient can be employed to formulate precise conditions that each of the W_s matrices must satisfy to fulfill their role of approximate decorrelation. To describe these conditions, we begin by introducing some convenient, special notation. For any matrix W, we let $rows(W)$ denote the number of rows in W. We denote the set of matrices having exactly n columns and no more than k rows by $\mathcal{M}_{k,n}$, or more typically by just $\mathcal{M}_k$ when there is no ambiguity about the value of n. Thus, for $W \in \mathcal{M}_k$, $rows(W) \leq k$. As a final bit of notation, we denote by $\mathcal{N}_{\gamma,s}$ the following set of matrices:

$$\mathcal{N}_{\gamma,s} \equiv \{W;\ \bar{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2}, \ldots, \chi_{s\alpha_q}, \chi_{s^c} \mid W\chi_s) \leq \gamma\}. \tag{3.19}$$

We see that any $W \in \mathcal{N}_{\gamma,s}$ leads to a state vector $x(s) = W\chi_s$ that fulfills the decorrelating role (2.3) to within a tolerance γ.

As we stated at the beginning of Section 3.3, there are two formulations of the multiscale realization problem that are of interest. For the first, we let γ_s denote the degree to which the state vector $x(s)$ should fulfill its decorrelating role (with $\gamma_s \in [0,1]$), and we seek a W_s satisfying

$$W_s = \arg \min_{W \in \mathcal{N}_{\gamma_s,s}} rows(W). \tag{3.20}$$

In the alternative formulation, let λ_s denote the maximum allowed dimension of the state vector $x(s)$, and we seek a W_s satisfying

$$W_s = \arg \min_{W \in \mathcal{M}_{\lambda_s}} \bar{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2}, \ldots, \chi_{s\alpha_q}, \chi_{s^c} \mid W\chi_s). \tag{3.21}$$

We refer to both (3.20) and (3.21) as versions of the *decorrelation* problem, where the objective is to (approximately) *decorrelate* the random vectors $\chi_{s\alpha_1}, \chi_{s\alpha_2}, \ldots, \chi_{s\alpha_q}$ and χ_{s^c}.

Clearly, our two formulations of the decorrelation problem are closely related. In fact, given an algorithm that yields a solution to one, that same algorithm can in principle be used to solve the other. To see this, let us suppose in particular we have an algorithm that solves (3.20). We can then iteratively apply this algorithm to find jointly the smallest real number γ^* in the set

$$\left\{\gamma \in [0,1];\ \mathcal{M}_{\lambda_s} \bigcap \mathcal{N}_{\gamma,s} \neq \emptyset\right\}, \tag{3.22}$$

and a matrix $W_s \in \mathcal{M}_{\lambda_s} \bigcap \mathcal{N}_{\gamma^*,s}$; the resulting matrix W_s is guaranteed to be an optimal solution to (3.21). This iterative solution to (3.21) has a very simple structure, in which we bracket the value γ^* by using the classical bisection method for root finding [57]. Our initial bracket for γ^* is the interval $[\gamma_-, \gamma_+]$, with $\gamma_- = 0$ and $\gamma_+ = 1$. At the i-th iteration, we then let γ be

$$\gamma \ = \ \frac{\gamma_+ + \gamma_-}{2},$$

and we solve for W as follows:

$$W \ = \ \arg \min_{W \in \mathcal{N}_{\gamma,s}} rows(W).$$

If $rows(W) > \lambda_s$, then we update $\gamma_- = \gamma$; otherwise, we update $\gamma_+ = \gamma$. Either way, we then proceed to the $i+1$-th iteration, continuing until the width of the bracketing interval $[\gamma_-, \gamma_+]$ is less than some prespecified tolerance ϵ. At that time, we terminate, and set $\gamma^* = \gamma_+$, which is guaranteed to be within ϵ of the actual minimum element of the set in (3.22).

Going the other way, suppose we have an algorithm that solves (3.21). We can then repeatedly apply this algorithm to find jointly the smallest non-negative integer k^* in the set

$$\left\{k \geq 0;\ \mathcal{M}_k \bigcap \mathcal{N}_{\gamma_s,s} \neq \emptyset\right\},$$

and a matrix $W_s \in \mathcal{M}_{k^*} \bigcap \mathcal{N}_{\gamma_s,s}$. This matrix is guaranteed to be an optimal solution to (3.20).

These iterative approaches for solving (3.20) and (3.21) highlight the intimate relationship between our two formulations of the problem. Furthermore, they will provide valuable insight as we carry out our algorithmic development in Section 3.4.

3.3.5 Summary

The multiscale stochastic realization problem has now been reduced to one of determining the W_s matrices, each parameterizing a state vector as in (3.7). For an exact realization, each W_s matrix must fulfill (3.8) exactly, or equivalently, must fulfill (3.20) with $\gamma_s = 0$. On the other hand, for a reduced-order realization, either (3.20) must be fulfilled for some choice of γ_s, or (3.21) must be fulfilled for some choice of λ_s.

Deferring until Section 3.4 the details of how the W_s matrices are found, to solve the decorrelation problem, we summarize as follows our algorithm for building a multiscale

model.

Algorithm 1 (*Overall modeling approach*)
Step 1. For $s = 0$ (i.e., the root node),
a) Determine W_0.
b) Set $P(0) = W_0 P_{\chi_0} W_0^T$.
Step 2. For $scale = 1.2, \ldots, M$,
a) For all elements of the set $\{s;\ m(s) = scale\}$,
i) Determine W_s.
ii) Set $P_{x(s)} = W_s P_{\chi_s} W_s^T$.
iii) Set $A(s) = W_s P_{\chi_s \chi_{s\bar{\gamma}}} W_{s\bar{\gamma}}^T P_{s\bar{\gamma}}^{-1}$.
iv) Set $B(s) = \left(P_{x(s)} - A(s) W_{s\bar{\gamma}} P_{\chi_{s\bar{\gamma}} \chi_s} W_s^T \right)^{1/2}$.

We can thus find all the model parameters in a single sweep from coarse to fine scales, determining W_s for each node as we go along, and thereby structuring the evolution of the multiscale dynamics so that the desired statistical behavior emerges at the finest scale.

The most attractive feature of this approach is its decomposition of the modeling problem into a collection of independent sub-problems that can each be solved myopically. We hasten to add, however, that by proceeding in this myopic fashion, we sacrifice tight control over the quality of the overall model fit. While in an informal sense, we certainly expect that there is a close relation between this overall quality of fit and the manner in which the W_s matrices are determined, this relationship is complicated, and is not at present well understood. For instance, it is unclear how fulfillment of (3.20) for some choice for γ_s relates to the loss in mean-square error (MSE) performance of the resulting model when it is used to carry out least-squares estimation; we expect that as γ_s is decreased, the loss in MSE performance will also decrease, but the precise nature of this relationship is not clear. Certainly, this is a topic in need of further research, as we discuss in greater detail in Section 6.2.1.

3.4 Solving the Decorrelation Problem

Now that we have a precisely stated modeling problem, we proceed to confront the heart of that problem, namely the determination of the W_s matrices. We begin by characterizing completely the optimal solutions to (3.20) and (3.21) for the special case of decorrelating a pair of random vectors. We then exploit these results to develop algorithms for decorrelating a whole collection of random vectors. These algorithms yield matrices W_s that can be applied directly in Algorithm 1 to obtain multiscale models.

3.4.1 Decorrelating a Pair of Random Vectors

As a first step toward solving the general decorrelation problem, we consider here the problem for the special case of decorrelating a pair of random vectors. Let us suppose η is a zero-mean random vector, having $(n_1 + n_2)$ components and covariance matrix P_η. We partition η into two sub-vectors, having respectively n_1 and n_2 components,

$$\eta = \begin{pmatrix} \eta_1^T & \eta_2^T \end{pmatrix}^T,$$

and we similarly partition the covariance matrix.

$$P_\eta \;=\; \begin{pmatrix} P_{\eta_1} & P_{\eta_1\eta_2} \\ P_{\eta_1\eta_2}^T & P_{\eta_2} \end{pmatrix}.$$

The specific problem we consider is that of decorrelating η_1 from η_2, where the decorrelating information is given by $W\eta$, for some matrix W that we must determine. In a manner analogous to (3.20) and (3.21), we consider two versions of the decorrelating problem. In the first, we let γ denote the degree to which the information $W\eta$ should fulfill its decorrelating role (with $\gamma \in [0,1]$), and we seek a matrix W satisfying

$$W \;=\; \arg \min_{W \in \mathcal{N}_\gamma} rows(W), \tag{3.23}$$

where $\mathcal{N}_\gamma$ denotes the following set of matrices:

$$\mathcal{N}_\gamma \;\equiv\; \{W;\; \bar{\rho}(\eta_1, \eta_2 \mid W\eta) \;\leq\; \gamma\}.$$

In the second version, we let λ denote the maximum allowed dimension of the decorrelating information $W\eta$, and we seek a matrix W satisfying

$$W \;=\; \arg \min_{W \in \mathcal{M}_\lambda} \bar{\rho}(\eta_1, \eta_2 \mid W\eta). \tag{3.24}$$

To supply a context for this problem, the random vectors η_1 and η_2 might represent, for instance, the left and right halves, respectively, of the finest scale of a multiscale process indexed on a dyadic tree (i.e., $\eta_1 = \chi_{0\alpha_1}$ and $\eta_2 = \chi_{0\alpha_2}$). In this case, both (3.23) and (3.24) are directly applicable to determining the information content of the root node of a dyadic-tree representation of χ_0. Alternatively, in the context of our discussion in Section 3.2 of time-series realization, η_1 (η_2) might correspond to η_{past} (η_{future}).

The following proposition provides a complete characterization of a matrix W that optimally solves (3.24). It turns out that the solution is closely related to the canonical correlation matrices for (η_1, η_2). We denote this triple of matrices by $(\hat{T}_1, \hat{T}_2, \hat{D})$, where $\hat{D} = \mathrm{diag}(d_1, d_2, \ldots, d_{m_{12}})$, with

$$1 \;\geq\; d_1 \;\geq\; d_2 \;\geq\; \cdots \;\geq\; d_{m_{12}}$$

and m_{12} denoting the rank of the cross-covariance $P_{\eta_1\eta_2}$.

Proposition 6 *For $0 \leq \lambda < m_{12}$ and for $i = 1, 2$,*

$$\min_{W \in \mathcal{M}_\lambda} \bar{\rho}(\eta_1, \eta_2 \mid W\eta) \;=\; \min_{W_i \in \mathcal{M}_\lambda} \bar{\rho}(\eta_1, \eta_2 \mid W_i\eta_i) \;=\; \bar{\rho}(\eta_1, \eta_2 \mid \hat{T}_{i,\lambda}\eta_i) \;=\; d_{\lambda+1}. \tag{3.25}$$

For $\lambda \geq m_{12}$,

$$\min_{W \in \mathcal{M}_\lambda} \bar{\rho}(\eta_1, \eta_2 \mid W\eta) \;=\; \min_{W_i \in \mathcal{M}_\lambda} \bar{\rho}(\eta_1, \eta_2 \mid W_i\eta_i) \;=\; \bar{\rho}(\eta_1, \eta_2 \mid \hat{T}_i\eta_i) \;=\; 0. \tag{3.26}$$

A complete, detailed proof is contained in Appendix B.

As a corollary, we have the following result, which provides a complete characterization of a matrix W that optimally solves (3.23). To prepare for a statement of the corollary, we

define (for $i = 1, 2$) $\mathcal{N}_{\gamma,i}$ to be the set of matrices

$$\mathcal{N}_{\gamma,i} \equiv \{W_i;\ \bar{\rho}(\eta_1, \eta_2 \mid W_i\eta_i) \leq \gamma\}. \tag{3.27}$$

Corollary 3 *For $\gamma \in [0,1]$,*

$$\min_{W \in \mathcal{N}_\gamma} rows(W) = \min_{W_i \in \mathcal{N}_{\gamma,i}} rows(W_i) = rows(\hat{T}_{1,i}) = j^*,$$

with $\hat{T}_{1,i} \in \mathcal{N}_{\gamma,i}$, and where j^* is such that

$$d_{j^*+1} \leq \gamma < d_{j^*} \tag{3.28}$$

(with $d_{m_{12}} + 1 \equiv 0$).

Proof: Let j^* satisfy (3.28). If $j^* = 0$, then we are done. If $j^* > 0$, then by Proposition 6,

$$\min_{W \in \mathcal{M}_{j^*-1}} \bar{\rho}(\eta_1, \eta_2 \mid W\eta) = d_{j^*} > \gamma. \tag{3.29}$$

But (3.29) implies that $rows(W) \geq j^*$ for $W \in M_\gamma$, and by combining this fact with the properties of $\hat{T}_{1,j^*}$, the corollary follows. **QED**.

3.4.2 Decorrelating a Collection of Random Vectors

Harnessing Propositions 5 and 6, as well as their corollaries, we now develop algorithms for addressing (3.20) and (3.21). We describe each algorithm, in turn, and then we examine their optimality.

Throughout, we denote by $(\hat{T}_{s\alpha_i}, \hat{T}_{(s\alpha_i)^c}, \hat{D}_{s,i})$ the canonical correlation matrices associated with $(\chi_{s\alpha_i}, \chi_{(s\alpha_i)^c})$, where the diagonal elements of $\hat{D}_{s,i}$ are denoted by $d^1_{s,i}, d^2_{s,i}, \ldots$. Also, we will find convenience in having special notation for the number of diagonal elements of $\hat{D}_{s,i}$ that are strictly greater than γ; we denote this number by $\sigma_{s,i}(\gamma)$:

$$\sigma_{s,i}(\gamma) \equiv \left|\left\{j;\ d^j_{s,i} > \gamma\right\}\right|.$$

Solving for W_s to address (3.20)

Our first algorithm yields a matrix $W_s \in \mathcal{N}_{\gamma_s,s}$, for any order tree $q \geq 2$. The construction of W_s is carried out in a sequence of q stages, where each stage consists of two steps: (i) determination of the information required to decorrelate a pair of random vectors as in Corollary 3, and (ii) incorporation of the resulting decorrelating information into the matrix W_s.

At the first stage of the procedure, we focus on decorrelating $\chi_{s\alpha_1}$ from $\chi_{(s\alpha_1)^c}$. To do so, we define $\mathcal{N}_{\gamma,1}$ as

$$\mathcal{N}_{\gamma,1} \equiv \left\{W;\ \bar{\rho}(\chi_{s\alpha_1}, \chi_{(s\alpha_1)^c} \mid W\chi_{s\alpha_1}) \leq \gamma\right\},$$

and then apply Corollary 3 to yield a matrix $W_{s,1}$ satisfying

$$W_{s,1} = \arg \min_{W \in \mathcal{N}_{\gamma_s,1}} rows(W).$$

Finally, we let

$$W_s^1 = \begin{pmatrix} W_{s,1} & \mathbf{0} \end{pmatrix},$$

where we pad with just enough zeros so that

$$\begin{aligned} W_s^1 \chi_s &= W_s^1 \begin{pmatrix} \chi_{s\alpha_1} \\ \chi_{s\alpha_2} \\ \vdots \\ \chi_{s\alpha_q} \end{pmatrix} \\ &= W_{s,1} \chi_{s\alpha_1}, \end{aligned}$$

and where the superscript 1 denotes that this matrix has been produced in the first stage of the procedure.

Let us now consider the i-th stage of the procedure, for $i \in \{2, 3, \ldots, q\}$, where we inductively assume that stages 1 through $i-1$ have already been executed. In a manner directly analogous to the first stage, we focus in the i-th stage on decorrelating $\chi_{s\alpha_i}$ from $\chi_{(s\alpha_i)^c}$. To do so, we define $\mathcal{N}_{\gamma,i}$ to be

$$\mathcal{N}_{\gamma,i} \equiv \left\{ W;\ \bar{\rho}(\chi_{s\alpha_i}, \chi_{(s\alpha_i)^c} \mid W\chi_{s\alpha_i}) \leq \gamma \right\}, \tag{3.30}$$

and then apply Corollary 3 to yield a matrix $W_{s,i}$ satisfying

$$W_{s,i} = \arg \min_{W \in \mathcal{N}_{\gamma_s,i}} rows(W). \tag{3.31}$$

Finally, we define W_s^i to be the block-diagonal matrix

$$W_s^i = \text{diag}(W_{s,1}, W_{s,2}, \ldots, W_{s,i}, \mathbf{0}), \tag{3.32}$$

where we pad with enough zeros so that

$$\begin{aligned} W_s^i \chi_s &= W_s^i \begin{pmatrix} \chi_{s\alpha_1} \\ \chi_{s\alpha_2} \\ \vdots \\ \chi_{s\alpha_q} \end{pmatrix} \\ &= \text{diag}(W_{s,1}, W_{s,2}, \ldots, W_{s,i}) \begin{pmatrix} \chi_{s\alpha_1} \\ \chi_{s\alpha_2} \\ \vdots \\ \chi_{s\alpha_i} \end{pmatrix}. \end{aligned}$$

Thanks to Corollary 3, the matrix W_s^i preserves all the decorrelating work done in the first $i-1$ stages of the procedure and captured in W_s^{i-1}. In particular, we are assured that for $j = 1, 2, \ldots, i$,

$$\bar{\rho}(\chi_{s\alpha_j}, \chi_{(s\alpha_j)^c} \mid W_s^i \chi_s) \leq \bar{\rho}(\chi_{s\alpha_j}, \chi_{(s\alpha_j)^c} \mid W_s^j \chi_s) \leq \bar{\rho}(\chi_{s\alpha_j}, \chi_{(s\alpha_j)^c} \mid W_{s,j} \chi_{s\alpha_j}) \leq \gamma_s,$$

and hence

$$\bar{\rho}(\chi_{s\alpha_1}, \ldots, \chi_{s\alpha_i}, \begin{pmatrix} \chi^T_{s\alpha_{i+1}} & \cdots & \chi^T_{s\alpha_q} & \chi^T_{s^c} \end{pmatrix}^T \mid W_s^i \chi_s) \;\leq\; \gamma_s.$$

Thus, since $W_s^q \in \mathcal{N}_{s,\gamma_s}$, we can complete the procedure by simply letting $W_s = W_s^q$.

The following is a summary of our algorithm for solving (3.20).

Algorithm 2 (*Solving for W_s to address (3.20)*)
Step 1. For $i = 1, 2, \ldots, q$,
a) Determine $(\hat{T}_{s\alpha_i}, \hat{T}_{(s\alpha_i)^c}, \hat{D}_{s,i})$ associated with $(\chi_{s\alpha_i}, \chi_{(s\alpha_i)^c})$.
b) Let j^* be the smallest integer for which $d^{j^*}_{s,i} \leq \gamma_s$.
c) Set $W_{s,i} = \hat{T}_{s\alpha_i,j^*}$.
Step 2. Set $W_s = \mathrm{diag}(W_{s,1}, W_{s,2}, \ldots, W_{s,q})$.

As this summary reveals, we do not ever actually use $\hat{T}_{(s\alpha_i)^c}$, and so in practice this matrix need not be computed. Thus, we can reduce computation by modifying Step 1a to the following:

Step 1a) Determine $(\hat{T}_{s\alpha_i}, \hat{D}_{s,i})$ associated with $(\chi_{s\alpha_i}, \chi_{(s\alpha_i)^c})$.

Solving for W_s to address (3.21)

In keeping with the discussion in Section 3.3.4, we use Algorithm 2 as the kernel of an iterative procedure for addressing (3.21). We summarize this approach as follows.

Algorithm 3 (*Solving for W_s to address (3.21)*)
If dimension$(\chi_s) \leq \lambda_s$, then set $W_s = I$.
Otherwise
Step 1. For $i = 1, 2, \ldots, q$, calculate $(\hat{T}_{s\alpha_i}, \hat{D}_{s,i})$ associated with $(\chi_{s\alpha_i}, \chi_{(s\alpha_i)^c})$.
Step 2. Find $\gamma_s^* = \min\{\gamma \in [0,1];\ \sum_{i=1}^q \sigma_{s,i}(\gamma) \leq \lambda_s\}$.
Step 3. For $i = 1, 2, \ldots, q$, set $W_{s,i} = T_{s\alpha_i,\sigma_{s,i}(\gamma_s^*)}$.
Step 4. Set $W_s = \mathrm{diag}(W_{s,1}, W_{s,2}, \ldots, W_{s,q})$.

We remark that if $q^{k-1} \leq \lambda_s \leq q^k$, $\forall s$ on a q-th order tree, then the **if** condition will hold for all the nodes at the k finest scales. On the other hand, for the coarser scales, the **otherwise** condition will hold. Also, we note that just as in Algorithm 2, there is no need to compute $\hat{T}_{(s\alpha_i)^c}$, thus explaining its absence in Step 1.

Optimality of Algorithms 2 and 3

Let us now consider the optimality of the two foregoing algorithms. We demonstrate that while Algorithm 2 is guaranteed to yield a matrix $W_s \in \mathcal{N}_{s,\gamma_s}$, there is no guarantee that W_s will have the minimum number of rows of any matrix in $\mathcal{M}_{s,\gamma_s}$. Thus, since Algorithm 3 is based on Algorithm 2, it too must be sub-optimal.

One of the primary sources of sub-optimality in Algorithm 2 is its restriction to block-diagonal matrices W_s. While our discussion in Section 3.3.3 indicated that relaxation of the block-diagonal form of W_s can lead to an increasing function

$$\bar{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2}, \ldots, \chi_{s\alpha_q}, \chi_{s^c},\ \mid W_s \chi_s)$$

as the amount of conditioning information increases, there is no guarantee that this increase will occur in general. In fact, to the contrary, it is fairly straightforward to concoct examples for which there exists a non-block-diagonal matrix $W \in \mathcal{M}_{\gamma_s,s}$ having fewer rows than the matrix W_s produced by our algorithm. As a specific example, let n_1, n_2, n_3, ν_1, ν_2, ν_3 be scalar, independent, identically distributed, zero mean, unit variance random variables, and suppose that we are building W_s for a particular node on a dyadic tree for which $\sigma = n_1 + n_2 + n_3$, with

$$\chi_{s\alpha_1} = \begin{pmatrix} \sigma + \nu_1 \\ n_1 \end{pmatrix}, \quad \chi_{s\alpha_2} = \begin{pmatrix} \sigma + \nu_2 \\ n_2 \end{pmatrix}, \quad \text{and} \quad \chi_{s^c} = \begin{pmatrix} \sigma + \nu_3 \\ n_3 \end{pmatrix},$$

If our objective is to choose W_s so that

$$\bar{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2}, \chi_{s^c} \mid W_s\chi_s) \leq 0.5, \tag{3.33}$$

then a bit of algebra will reveal that by letting W_s be such that $W_s\chi_s = \sigma$ (requiring that W_s have only a single row that is in violation of the block-diagonal restriction), then (3.33) can be achieved with equality. On the other hand, the symmetry of this particular problem implies that Algorithm 2 will yield a matrix W_s having *at least* three rows, in order to fulfill (3.33), which is clearly sub-optimal.

Even within the class of block-diagonal matrices, Algorithm 2 is not guaranteed to find a minimal matrix. One reason is that the decorrelating information generated in each of the q stages is determined without regard for the information generated in the other stages. This problem can be partially mitigated by making a single modification to our original algorithm. Specifically, we modify the definition of $\mathcal{N}_{\gamma,i}$ from the one given in (3.30) to the following alternative:

$$\mathcal{N}_{\gamma,i} \equiv \left\{ W;\ \bar{\rho}(\chi_{s\alpha_i}, \chi_{(s\alpha_i)^c} \mid W_s^{i-1}\chi_s, W\chi_{s\alpha_i}) \leq \gamma \right\}. \tag{3.34}$$

We thereby obtain, at stage i of the procedure, a matrix $W_{s,i}$ which takes better account of the decorrelating information generated in the $i-1$ preceding stages. As a result, the number of rows in $W_{s,i}$ is non-increasing as (3.30) is replaced by (3.34), and so the number of rows in the final W_s is also non-increasing as (3.30) is replaced by (3.34). However, a considerable price in computational complexity must be paid to use this modified form of Algorithm 2, since we must calculate conditioned covariance matrices at each of the q stages.

Moreover, while this modification, in which we replace (3.30) with (3.34), will certainly improve performance, we are still not guaranteed to find a minimal matrix, even within the class of block-diagonal ones. In fact, the sequential nature of the stages of the algorithm is such that the output matrix W_s is highly dependent on the ordering of the block components of χ_0,

$$\chi_0^T = \begin{pmatrix} \chi_{s\alpha_1}^T & \chi_{s\alpha_2}^T & \cdots & \chi_{s\alpha_q}^T & \chi_{s^c}^T \end{pmatrix}^T.$$

To clarify this remark, let $\mathcal{P}$ be any matrix that permutes the block components of χ_0, for instance, $\mathcal{P}$ might be such that

$$(\mathcal{P}\chi_0)^T = \begin{pmatrix} \chi_{s\alpha_q}^T & \chi_{s\alpha_{q-1}}^T & \cdots & \chi_{s\alpha_1}^T & \chi_{s^c}^T \end{pmatrix}^T.$$

If we then apply our algorithm to the random vector $\mathcal{P}\chi_0$, we will obtain a matrix W_s' for which

$$\tilde{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2}, \ldots, \chi_{s\alpha_q}, \chi_{s^c} \mid W_s'\mathcal{P}\chi_s) \quad \leq \quad \gamma_s,$$

and hence, $\mathcal{P}W_s' \in \mathcal{N}_{\gamma_s,s}$. The point here is that W_s may be unequal to $\mathcal{P}W_s'$, and the latter matrix may very well have fewer rows than the former.

To construct a specific example, let n, ν_1, ν_2, ν_3 and ν_4 be scalar, independent, identically distributed, zero mean, unit variance random variables, and suppose that we are building W_s for a particular node on a dyadic tree for which

$$\chi_{s\alpha_1} \quad = \quad \begin{pmatrix} n+\nu_1 \\ n+\nu_2 \end{pmatrix}, \quad \chi_{s\alpha_2} \quad = \quad n \quad \text{and} \quad \chi_{s^c} \quad = \quad \begin{pmatrix} n+\nu_3 \\ n+\nu_4 \end{pmatrix}$$

By inspection, if we let W_s be such that $W_s\chi_s = \chi_{s\alpha_2}$ (requiring that W_s have only a single row), then

$$\tilde{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2}, \chi_{s^c} \mid W_s\chi_s) \quad = \quad 0.$$

On the other hand, because of the ordering of the stages of our algorithm, the output will be a matrix having *two* rows, for $\gamma_s = 0$.

To the best of our knowledge, the only way to obtain a matrix W_s that is insensitive to this permutation issue is to try all block permutations, and use the one yielding the minimum number of rows.

3.4.3 Calculating the Canonical Correlation Matrices

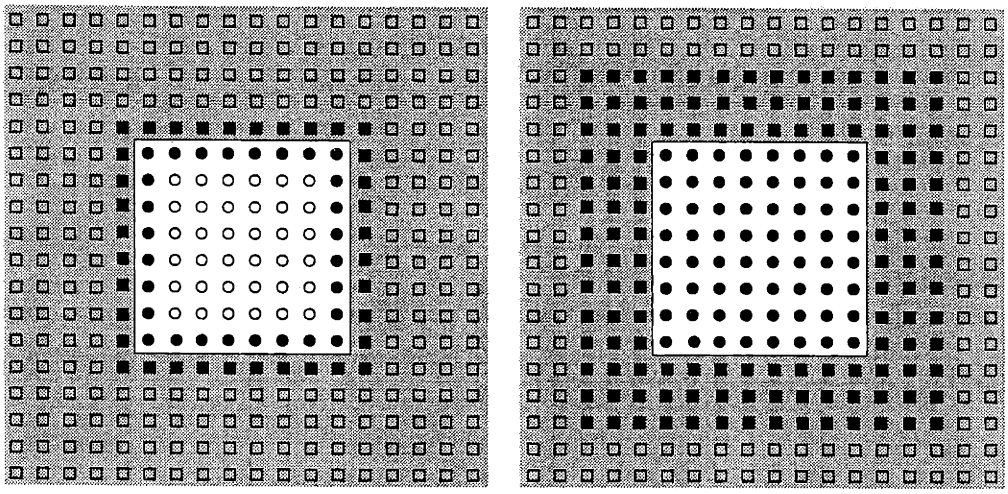
At the core of Algorithms 2 and 3 is the need to calculate the canonical correlation matrices $(\hat{T}_s, \hat{D}_s)$ associated with pairs of random vectors of the form (χ_s, χ_{s^c}). From Section 2.4.3, we know that the calculations associated with a single pair (χ_s, χ_{s^c}) require roughly $\mathcal{O}(N^3)$ floating point operations, where N is the dimension χ_{s^c}. Thus, for problems of practical interest to us, the exact calculation of these matrices is generally out of the question. For instance, if χ_0 represents a random field of dimension 256×256, then N will be roughly 5×10^4.

For Markov random processes and fields, this computational load can be reduced considerably, by properly exploiting Proposition 4. In particular, even if χ_0 represents a WSMRF as large as 256×256, then the canonical correlation matrices associated with (χ_s, χ_{s^c}) can be computed in a manageable fashion to machine precision. Moreover, for non-Markov processes and fields, a slight generalization of this approach serves effectively as a method for obtaining good approximate results.

We illustrate the approach by considering, in parallel, a 1-D example and a 2-D example. In our 1-D example, we let χ_0 represent the values of a first-order, scalar-valued WS Markov process over a segment of length 256^2, while in our 2-D example, we let χ_0 represent the values of a first-order, scalar-valued WSMRF over a 256×256 square region of the plane. We focus on a particular node s for which χ_s and χ_{s^c} contain the values of the process (field) at the subsets of points displayed in Figure 3-1a (Figure 3-2a). Specifically, χ_s contains the values of the process (field) at the 8 (64) grid points marked with circles, both filled and not filled, in the white region, while χ_{s^c} contains the values at the all other grid points; subsets of these other grid points are displayed in the figures, where they are marked with

(a) (b)

Figure 3-1: Illustration of our approach to finding the canonical correlation matrices associated with (χ_s, χ_{s^c}) for (a) a first-order WS Markov process, and (b) a non-Markov random process. In both cases, the vector χ_s contains the field values at the 8 grid points marked with circles (both filled and not filled) in the white region, while χ_{s^c} contains the values at the other grid points, all marked with squares (both filled and not filled). Also in both cases, the vector μ_s contains the subset of χ_s values at the grid points marked with filled-in circles, while μ_{s^c} contains the subset of χ_{s^c} values at the grid points marked with filled-in squares. The dimension of both μ_s and μ_{s^c} is lower in (a) than in (b), owing to the WS reciprocal nature of the process in (a).



(a) (b)

Figure 3-2: Illustration of our approach to finding the canonical correlation matrices associated with (χ_s, χ_{s^c}) for (a) a first-order WSMRF, and (b) a non-Markov random field. In both cases, the vector χ_s contains the field values at the 64 grid points marked with circles (both filled and not filled) in the white region, while χ_{s^c} contains the values at the other grid points, all marked with squares (both filled and not filled). Also in both cases, the vector μ_s contains the subset of χ_s values at the grid points marked with filled-in circles, while μ_{s^c} contains the subset of χ_{s^c} values at the grid points marked with filled-in squares. The dimension of both μ_s and μ_{s^c} is lower in (a) than in (b), owing to the WS Markov property of the field in (a).

squares, both filled and not filled.

Let us consider how Proposition 4 can be exploited to simplify the calculation of the canonical correlation matrices associated with (χ_s, χ_{s^c}). To use Proposition 4, we must devise selection matrices Θ_s and Θ_{s^c} such that the random vectors μ_s and μ_{s^c},

$$\mu_s = \Theta_s \chi_s \quad \text{and} \quad \mu_{s^c} = \Theta_{s^c} \chi_{s^c}, \tag{3.35}$$

capture the correlated component of χ_s and χ_{s^c}, in the sense that

$$E\left[(\chi_s - E(\chi_s|\mu))\,(\chi_{s^c} - E(\chi_{s^c}|\mu))^T\right] = \mathbf{0}, \tag{3.36}$$

for both $\mu = \mu_s$ *and* $\mu = \mu_{s^c}$. Thanks to the Markov property, this task can be carried out by inspection for both of our examples. In both cases, we simply let Θ_s and Θ_{s^c} be selection

matrices[3] chosen such that μ_s and μ_{s^c} contain the values of χ_s and χ_{s^c} at their respective boundary points, where these boundary points are marked in the figures with filled-in circles and squares, respectively. If we denote by $(\hat{T}_s^\mu, \hat{T}_{s^c}^\mu, \hat{D}_s^\mu)$ the canonical correlation matrices associated with (μ_s, μ_{s^c}), then thanks to Proposition 4,

$$\hat{T}_s = \hat{T}_s^\mu \Theta_s, \quad \text{and} \quad \hat{D}_s = \hat{D}_s^\mu. \tag{3.37}$$

To see the computational savings that can result, we note that in our 2-D example, the dimension of μ_{s^c} is roughly 5×10^{-4} times the dimension of χ_{s^c}, this approach reduces the computational cost of determining $(\hat{T}_s, \hat{D}_s)$ by roughly a factor of 6×10^9. From this example, the structure of our approach should be clear, for any case in which we are modeling a WS Markov random process or field.

For non-Markov random processes and fields, there is no guarantee that the correlated component of (χ_s, χ_{s^c}) can be captured by boundary information over a region as thin as the one used in our foregoing examples. To compensate for this fact, we modify our approach slightly for the non-Markov case. Our modified strategy is to make the boundary region as thick as possible for each of χ_s and χ_{s^c}, subject to the constraint that the resulting vectors μ_s and μ_{s^c} have dimension no greater than some prescribed limit. Using the same graphical conventions as in Figures 3-1a and 3-2a, this idea is illustrated in part b of the respective figures; in Figure 3-1b, 6 is the limiting dimension of both μ_s and μ_{s^c}, while in Figure 3-2b, 132 is the limiting dimension of both μ_s and μ_{s^c}. Once μ_s and μ_{s^c} have been defined, we proceed exactly as in the Markov case.

Overall, our approach for both the Markov and non-Markov cases can be summarized as follows:

Algorithm 4 (*Calculation of canonical correlation matrices*)
Step 1. Impose an upper bound θ_{rows} on the number of rows in Θ_s and Θ_{s^c}.
Step 2. Define selection matrices Θ_s and Θ_{s^c}.
a) Make boundary regions as thick as possible, with up to θ_{rows} in each of Θ_s and Θ_{s^c}.
b) Set $\mu_s = \Theta_s \chi_s$ and $\mu_{s^c} = \Theta_{s^c} \chi_{s^c}$.
Step 3. Determine $(\hat{T}_s^\mu, \hat{D}_s^\mu)$ associated with (μ_s, μ_{s^c}).
Step 4. Define $(\hat{T}_s, \hat{D}_s)$ as in (3.37).

We remark that the selection matrices Θ_s are quite sparse, and thus, the matrix multiplication involved in the definition of $\hat{T}_s$ in (3.37) should be computed with care, taking advantage of this sparsity. This issue is discussed in detail, in the next section, where we summarize our overall modeling algorithm.

3.5 Summary of Modeling Algorithms

Taken together, Algorithms 1-4 contain all the details of our approach to multiscale modeling. Here we reassemble those details into a single, coherent package. We then describe the considerable simplifications that are possible when χ_0 represents a process or field that is WS stationary.

[3] A *selection* matrix consists solely of zeros and ones, with the additional restriction that each row have exactly one non-zero component and each column have at most one non-zero component.

3.5.1 General Algorithm–No Stationarity Assumption

Algorithm 1′ (*Overall modeling approach, comprehensive summary*)
Step 1. For $s = 0$ (i.e., the root node),
a) Determine W_0, using either Algorithm 2′ or 3′.
b) Set $P(0) = W_0 P_{\chi_0} W_0^T$.
Step 2. For $scale = 1, 2, \ldots, M$,
a) For all elements of the set $\{s;\ m(s) = scale\}$,
i) Determine W_s, using either Algorithm 2′ or 3′.
ii) Set $P_{x(s)} = W_s P_{\chi_s} W_s^T$.
iii) Set $A(s) = W_s P_{\chi_s \chi_{s\bar{\gamma}}} W_{s\bar{\gamma}}^T P_{s\bar{\gamma}}^{-1}$.
iv) Set $B(s) = \left(P_{x(s)} - A(s) W_{s\bar{\gamma}} P_{\chi_{s\bar{\gamma}} \chi_s} W_s^T\right)^{1/2}$.

Calculating W_s matrices

Algorithm 2′ (*Solving for W_s to address (3.20), comprehensive summary*)
Step 1. For $i = 1, 2, \ldots, q$,
a) Define $(\Theta_{s\alpha_i}, \Theta_{(s\alpha_i)^c})$ as appropriate selection matrices.
b) Set $\mu_{s\alpha_i} = \Theta_{s\alpha_i} \chi_{s\alpha_i}$ and $\mu_{(s\alpha_i)^c} = \Theta_{(s\alpha_i)^c} \chi_{(s\alpha_i)^c}$.
c) Determine $(\hat{T}^{\mu}_{s\alpha_i}, \hat{D}^{\mu}_{s,i})$ associated with $(\mu_{s\alpha_i}, \mu_{(s\alpha_i)^c})$.
d) Set $\hat{T}_{s\alpha_i} = \hat{T}^{\mu}_{s\alpha_i} \Theta_{s\alpha_i}$ and $\hat{D}_{s,i} = \hat{D}^{\mu}_{s,i}$.
e) Let j^* be the smallest integer for which $d^{j^*}_{s,i} \leq \gamma_s$.
f) Set $W_{s,i} = \hat{T}_{s\alpha_i, j^*}$.
Step 2. Set $W_s = \mathrm{diag}(W_{s,1}, W_{s,2}, \ldots, W_{s,q})$.

Algorithm 3′ (*Solving for W_s to address (3.21), comprehensive summary*)
If dimension$(\chi_s) \leq \lambda_s$, then set $W_s = I$.
Otherwise
Step 1. For $i = 1, 2, \ldots, q$,
(a) -(d) Carry out steps 1(a), (b), (c) and (d) of Algorithm 2′, without modification.
e) Define $\sigma_{s,i}(\gamma) \equiv \left|\left\{j;\ d^j_{s,i} > \gamma\right\}\right|$.
Steps 2-4. Carry out steps 2, 3 and 4 of Algorithm 3, in Section 3.4.2, without modification.

The computational complexity of both Algorithms 2′ and 3′ is $\mathcal{O}(\theta^3_{rows_s})$, where θ_{rows} denotes our imposed upper-bound on the number of rows in each the selection matrix Θ_s.

Calculating $P(0)$, $A(s)$ and $B(s)$

Let us now consider how we exploit the sparsity of the W_s matrices to calculate $A(s)$ and $B(s)$ and $P_{x(s)}$, beginning with consideration of $P_{x(s)}$. Towards this end, we note that the

covariance P_{χ_s} can be block-decomposed as

$$P_{\chi_s} = \begin{pmatrix} P_{\chi_{s\alpha_1}} & P_{\chi_{s\alpha_1}\chi_{s\alpha_2}} & \cdots & P_{\chi_{s\alpha_1}\chi_{s\alpha_q}} \\ P_{\chi_{s\alpha_2}\chi_{s\alpha_1}} & P_{\chi_{s\alpha_2}} & \cdots & P_{\chi_{s\alpha_2}\chi_{s\alpha_q}} \\ \vdots & \vdots & \ddots & \vdots \\ P_{\chi_{s\alpha_q}\chi_{s\alpha_1}} & P_{\chi_{s\alpha_q}\chi_{s\alpha_2}} & \cdots & P_{\chi_{s\alpha_q}} \end{pmatrix},$$

and hence, in light of the block-diagonal structure of W_s, $P_{x(s)}$ can be block-decomposed in the following way:

$$\begin{aligned} P_{x(s)} &= W_s P_{\chi_s} W_s^T \\ &= \begin{pmatrix} W_{s,1}P_{\chi_{s\alpha_1}}W_{s,1}^T & W_{s,1}P_{\chi_{s\alpha_1}\chi_{s\alpha_2}}W_{s,2}^T & \cdots & W_{s,1}P_{\chi_{s\alpha_1}\chi_{s\alpha_q}}W_{s,q}^T \\ W_{s,2}P_{\chi_{s\alpha_2}\chi_{s\alpha_1}}W_{s,1}^T & W_{s,2}P_{\chi_{s\alpha_2}}W_{s,2}^T & \cdots & W_{s,2}P_{\chi_{s\alpha_2}\chi_{s\alpha_q}}W_{s,q}^T \\ \vdots & \vdots & \ddots & \vdots \\ W_{s,q}P_{\chi_{s\alpha_q}\chi_{s\alpha_1}}W_{s,1}^T & W_{s,q}P_{\chi_{s\alpha_q}\chi_{s\alpha_2}}W_{s,2}^T & \cdots & W_{s,q}P_{\chi_{s\alpha_q}}W_{s,q}^T \end{pmatrix}. \end{aligned}$$

Using now the definition of $W_{s,i}$, we see that each block component of $P_{x(s)}$ can be decomposed as

$$W_{s,i}P_{\chi_{s\alpha_i}\chi_{s\alpha_j}}W_{s,j}^T = \hat{T}_{s\alpha_i,k_i}\left(\Theta_{s\alpha_i}P_{\chi_{s\alpha_i}\chi_{s\alpha_j}}\Theta_{s\alpha_j}^T\right)\hat{T}_{s\alpha_j,k_j}^T, \tag{3.38}$$

where k_i and k_j are appropriate integers (whose values follow from the particular algorithm used to determine the matrices $W_{s,i}$ and $W_{s,j}$). The important point here is that because $\Theta_{s\alpha_i}$ and $\Theta_{s\alpha_j}$ are selection matrices, the calculation of matrix products of the form $\Theta_{s\alpha_i}P_{\chi_{s\alpha_i}\chi_{s\alpha_j}}\Theta_{s\alpha_j}^T$ involves nothing more than selecting elements from the matrix $P_{\chi_{s\alpha_i}\chi_{s\alpha_j}}$; since we select θ_{rows}^2 elements, the complexity of calculating these matrix products is $\mathcal{O}(\theta_{rows}^2)$, *independent* of the dimension of $P_{\chi_{s\alpha_i}\chi_{s\alpha_j}}$. Thus, the overall computational complexity of calculating (3.38) is upper-bounded by $\mathcal{O}(\theta_{rows}^3)$, and since the complexity of calculating $P_{x(s)}$ is roughly q^2 times the effort required to calculate any single block of $P_{x(s)}$, it follows that the total complexity of calculating $P_{x(s)}$ is upper-bounded by $\mathcal{O}(q^2\theta_{rows}^3)$.

By a very similar line of reasoning, it follows that the calculation of $A(s)$ and $B(s)$ can also be carried out with complexity $\mathcal{O}(q^2\theta_{rows}^3)$. We omit the details.

Overall computational complexity

The computational complexity of this model-building algorithm is $\mathcal{O}(\theta_{rows}^3)$ per tree node. Letting N be the number of finest-scale nodes, it thus follows that the overall complexity is $\mathcal{O}(N\theta_{rows}^3)$.

3.5.2 Specialized Algorithm–Stationarity Assumption

When the random process or field to be modeled is WS stationary (but not necessarily WS Markov), we can streamline Algorithms 1, 2′ and 3′. The most immediate effect of stationarity is that the covariance matrix P_{χ_s} becomes independent of position within any fixed scale. Also, the cross-covariance matrix $P_{\chi_s\chi_{s\bar{\gamma}}}$ takes on only q distinct values at any fixed scale. In particular, at a fixed scale, every node s that is the i-th child of its parent node $s\bar{\gamma}$ (so that $s = s\bar{\gamma}\alpha_i$) shares the same value for $P_{\chi_s\chi_{s\bar{\gamma}}}$, for $i = 1, 2, \ldots, q$.

Finally, and perhaps most importantly, stationarity can be exploited to develop and justify an approximation scheme in which the matrix W_s becomes *independent* of the location of the node s within any fixed scale. By an abuse of notation, we thus write $W_s = W_{m(s)}$, where we defer until later in this section the details and justification of this approximation.

Let us consider the effect of stationarity on Algorithm 1. For one, the covariance matrix $P_{x(s)}$ satisfies $P_{x(s)} = P_{m(s)}$. Also, the matrices $A(s)$ and $B(s)$ can take on only q distinct values for each scale, where the i-th value is taken, for $i = 1, 2, \ldots, q$, if node s is the i-th child of node $s\bar{\gamma}$. We thus again abuse notation by writing $A(m(s), i)$ and $B(m(s), i)$ rather than $A(s)$ and $B(s)$, where i is the unique element of $\{1, 2, \ldots, q\}$ for which $s = s\bar{\gamma}\alpha_i$. With these simplifications established, the following algorithm is used in lieu of Algorithm 1.

Algorithm 5 (*Overall modeling approach, stationary case*)
Step 1. For $s = 0$ (i.e., the root node),
a) Determine W_0, using either Algorithm 6 or 7.
b) Set $P(0) = W_0 P_{\chi_0} W_0^T$.
Step 2. For $m = 1, 2, \ldots, M$,
a) Determine W_m, using either Algorithm 6 or 7.
b) Set $P_m = W_m P_{\chi_s} W_m^T$, using any node s at scale m.
c) For $i = 1, 2, \ldots, q$,
i) Select a node s at scale m such that $s = s\bar{\gamma}\alpha_i$.
ii) Set $A(m,i) = W_m P_{\chi_s \chi_{s\bar{\gamma}}} W_{m-1}^T P_{m-1}^{-1}$.
iii) Set $B(m,i) = \left(P_m - A(m,i) W_{m-1} P_{\chi_{s\bar{\gamma}} \chi_s} W_m^T\right)^{1/2}$.

Calculating W_m

One might intuitively expect that when stationarity holds, the W_s matrices *automatically* satisfy $W_s = W_{m(s)}$. The difficulty, though, is that we are only realizing processes and fields over finite sets of grid points, and so boundary effects can arise that disrupt the offset invariance of the W_s matrices. To see this fact, we examine our relations (3.20) and (3.21) for determining the W_s matrices. Both of these relations are closely tied to the value of the following generalized correlation coefficient:

$$\bar{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2}, \cdots, \chi_{s\alpha_q}, \chi_{s^c} \mid W\chi_s). \tag{3.39}$$

While stationarity assures us that

$$\bar{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2}, \cdots, \chi_{s\alpha_q} \mid W\chi_s) \;=\; \bar{\rho}(\chi_{\sigma\alpha_1}, \chi_{\sigma\alpha_2}, \cdots, \chi_{\sigma\alpha_q} \mid W\chi_\sigma)$$

for any two nodes s and σ at a common scale, the finiteness of χ_0 implies that the statistical relationship between χ_s and χ_{s^c} will almost always be different from the statistical relationship between χ_σ and χ_{σ^c}, and as a consequence of this difference,

$$\bar{\rho}(\chi_{s\alpha_1}, \chi_{s\alpha_2} \cdots, \chi_{s\alpha_q}, \chi_{s^c} \mid W\chi_s) \;\neq\; \bar{\rho}(\chi_{\sigma\alpha_1}, \chi_{\sigma\alpha_2}, \cdots, \chi_{\sigma\alpha_q}, \chi_{\sigma^c} \mid W\chi_\sigma).$$

The implication of this last inequality is that we must modify our relations (3.20) and (3.21) in order to achieve $W_s = W_{m(s)}$. As we now show, the needed modification is straightforward. The key is to embed the vectors χ_0, χ_s and χ_{s^c} in an larger vector $\bar{\chi}_0$ that contains the values of the random process or field (to be modeled) over its entire, possibly

infinite, extent; actually, the only case in which $\bar{\chi}_0$ will be finite is when the random process or field is indexed on a circle or toroid, respectively. With χ_0 embedded in $\bar{\chi}_0$, we can readily do away with all boundary effects. To proceed, we define $\bar{\chi}_{s^c}$ to be the particular sub-vector of $\bar{\chi}_0$ that contains all the values of $\bar{\chi}_0$ that are not in the vector χ_s; in other words, $\bar{\chi}_{s^c}$ contains χ_{s^c} as a sub-vector, and is also such that

$$\bar{\chi}_0 \;=\; \mathcal{P}_s \begin{pmatrix} \chi_s \\ \bar{\chi}_{s^c} \end{pmatrix},$$

for some permutation matrix $\mathcal{P}_s$.

For the purposes of determining the W_s matrices, we now simply replace χ_{s^c} with $\bar{\chi}_{s^c}$ in (3.19)-(3.21). The immediate result is that $W_s = W_{m(s)}$, so long of course as $\lambda_s = \lambda_{m(s)}$ or $\gamma_s = \gamma_{m(s)}$. There is a tradeoff involved in this replacement strategy. On the one hand, our decorrelation task is made more difficult with the replacement, because now each of the vectors $\chi_{s\alpha_1}, \chi_{s\alpha_2}, \ldots, \chi_{s\alpha_q}$ must not only be decorrelated from χ_{s^c}, but also from the larger vector $\bar{\chi}_{s^c}$; we are effectively being more conservative than we have to be, as we determine the amount of decorrelating information to include in $W_s\chi_s$. On the other hand, by being more conservative, we achieve the relation $W_s = W_{m(s)}$, whose simplifying effect has already been seen on both the overall flow of our modeling algorithm, and on the symmetry of the resulting parameters $A(s)$ and $B(s)$. Moreover, the new matrix W_s is more intrinsically tied to χ_s than the old W_s, in the sense that the new matrix is independent of the size of the finite subset of $\bar{\chi}_0$ for which we are building a multiscale model. Thus, if we later decide to build a multiscale model for a larger region of the process or field $\bar{\chi}_0$, then we can still use the same W_s matrix already computed.

We now describe our modified algorithms for determining the W_s matrices. An important observation in this regard is that the canonical correlation matrices $(\hat{T}_s, \hat{D}_s)$ associated with $(\chi_s, \bar{\chi}_{s^c})$ satisfy $\hat{T}_s = \hat{T}_{m(s)}$ and $\hat{D}_s = \hat{D}_{m(s)}$.

When addressing (3.20), the following algorithm is used in lieu of Algorithm 2′ to calculate W_m.

Algorithm 6 (*Solving for W_m to address (3.20), stationary case*)
Step 1. Let s be any node at scale m; let $\sigma = s\alpha_1$.
Step 2. Define $(\Theta_\sigma, \Theta_{\sigma^c})$ as appropriate selection matrices.
Step 3. Set $\mu_\sigma = \Theta_\sigma \chi_\sigma$ and $\mu_{\sigma^c} = \Theta_{\sigma^c} \bar{\chi}_{\sigma^c}$.
Step 4. Calculate $(\hat{T}^{\mu}_{m+1}, \hat{D}^{\mu}_{m+1})$ associated with $(\mu_\sigma, \mu_{\sigma^c})$.
Step 5. Set $\hat{T}_{m+1} = \hat{T}^{\mu}_{m+1}$ and $\hat{D}_{m+1} = \hat{D}^{\mu}_{m+1}$.
Step 6. Let j^* be the smallest integer for which $d^{j^*}_{m+1} \leq \gamma_m$.
Step 7. Set W_m be a block-diagonal matrix, having q blocks, each equal to $\hat{T}_{m+1,j^*}$.

On the other hand, when addressing (3.21), the following algorithm is used in lieu of Algorithm 3′ to calculate W_m. Symmetry here tells us that we should devote an equal number of components of the state vector $x(s)$ to decorrelating each of $\chi_{s\alpha_1}$, $chi_{s\alpha_2}, \ldots,$ $\chi_{s\alpha_q}$ from $\chi_{(s\alpha_1)^c}, \chi_{(s\alpha_2)^c}, \ldots, \chi_{(s\alpha_q)^c}$, respectively; hence, λ_m should be set to a multiple of

q.

Algorithm 7 (*Solving for W_m to address (3.21), stationary case*)
If dimension$(\chi_s) \leq \lambda_s$, then set $W_m = I$.
Otherwise
Steps 1-4. Carry out steps 1-4 of Algorithm 6, without modification.
Step 5. Let W_m be a block-diagonal matrix, having q blocks, each equal to $\hat{T}_{m+1,\lambda_m/q}$.

The computational complexity of both Algorithms 6 and 7 is $\mathcal{O}(\theta_{rows}^3)$, where θ_{rows} denotes our imposed upper-bound on the number of rows in each the selection matrix Θ_m.

Overall computational complexity

In contrast to our general modeling algorithm, the computational complexity of this specialized algorithm is $\mathcal{O}(\theta_{rows}^3)$ per *scale*. Hence, the overall computational complexity is $\mathcal{O}(M\theta_{rows}^3)$.

3.6 Application of the Model-Building Algorithms

In this section, we present four multiscale modeling examples that display the generality and effectiveness of our modeling procedure.

3.6.1 WS Stationary Random Process Having a Damped-Sinusoid Correlation Function

For our first example, we consider a stationary 1-D random process y_n whose correlation function $R_{yy}(\cdot)$ is a damped sinusoid:

$$R_{yy}(m) \;=\; E\,(y(n)y(n-m)) \;=\; e^{-\alpha|m|}\cos(\omega_0 m),$$

having damping factor α and frequency ω_0. A plot of this correlation function for parameter values

$$\alpha \;=\; 0.01 \quad \text{and} \quad \omega \;=\; \frac{2\pi}{127/5} \tag{3.40}$$

is represented by the solid curve in Figure 3-3. We remark that damped, oscillatory correlation functions are frequently encountered in practice; for example, such a function has been successfully used to model the correlation of fading radio signals [67]. Our reason for considering this particular correlation function is that it exhibits long-range, periodic structure, which we expect to pose a most stringent challenge to our modeling approach.

We consider the problem of building a multiscale model, indexed on a dyadic tree, to realize a 128-point segment of this process. To assign values to the parameters of $R_{yy}(\cdot)$, we use (3.40). We constrain the multiscale model dimension, and then apply Algorithms 5 and 7, with the parameter θ_{rows} set to 128.

Preliminary analysis: value to use for dimension constraint

Let us begin with some preliminary analysis to determine a reasonable range of values to use for our constraint on multiscale model dimension λ_m. Although this analysis is not, strictly speaking, necessary to apply our algorithm, it is nevertheless useful, and it makes some nice ties back to our examples in Section 2.3 on realizing on Markov processes and fields.

The spectral density $S_{yy}(\cdot)$ corresponding to $R_{yy}(\cdot)$ is obtained as follows:

$$\begin{aligned} S_{yy}(z) &= \sum_{n=-\infty}^{\infty} R_{yy}(n) z^{-n} \\ &= \frac{1 - |\beta|^4 + \gamma(z + z^{-1})}{(1 - \beta z^{-1})(1 - \beta^* z^{-1})(1 - \beta z)(1 - \beta^* z)} \end{aligned} \tag{3.41}$$

with

$$\beta \equiv e^{-(\alpha - \sqrt{-1}\omega_0)}, \quad \gamma = \frac{1}{2}(\beta + \beta^*)\left(|\beta|^2 - 1\right).$$

We can factor (3.41) as

$$S_{yy}(z) = H(z)H(z^{-1}),$$

where, for the correlation parameter values given in (3.40),

$$H(z) \approx \frac{-0.1562(1 - 0.7787z^{-1})}{(1 - \beta z^{-1})(1 - \beta^* z^{-1})} \approx \frac{-0.1562(1 - 0.7787z^{-1})}{1 - 1.9198z^{-1} + 0.9802z^{-2}} \tag{3.42}$$

This transfer function is causal, stable and indicates that y_n can be realized by a second-order state-space model, driven by white noise, exactly as in (3.2). For example, one possible choice for the state-space parameters is as follows:

$$A = \begin{pmatrix} 1.9198 & -0.9802 \\ 1 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0.7944 \end{pmatrix}, \quad c = \begin{pmatrix} -0.1562 & 0 \end{pmatrix}. \tag{3.43}$$

Now that we have discerned the Markov structure of the process y_n, it follows readily that there must exist a low-order, exact multiscale representation for y_n. In particular, by closely following the construction of the models in the examples of Section 2.3, we can build an exact realization in which the state vectors have a dimension no greater than 8, with each state vector containing appropriate boundary information. At the root node, the state is defined as

$$x(0) = \begin{pmatrix} z^T(0) & z^T(63) & z^T(64) & z^T(127) \end{pmatrix}^T,$$

while at the two children of the root node, the state are defined as

$$\begin{aligned} x(0\alpha_1) &= \begin{pmatrix} z^T(0) & z^T(31) & z^T(32) & z^T(63) \end{pmatrix}^T \\ x(0\alpha_2) &= \begin{pmatrix} z^T(64) & z^T(95) & z^T(96) & z^T(127) \end{pmatrix}^T. \end{aligned} \tag{3.44}$$

Scale after scale, we continue in this manner to divide the process into conditionally in-

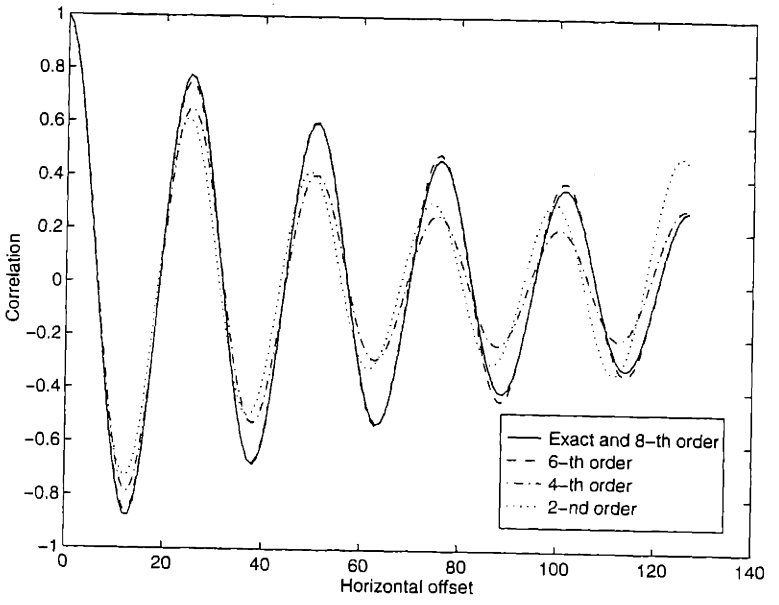


Figure 3-3: Comparison of the fidelity of four multiscale models for representing a stationary random process having an oscillatory correlation function. The solid curve displays the desired correlation function, which can be realized exactly with an 8-th order multiscale model. The correlation function of the 6-th order model is very close to the exact, while there is more noticeable degradation in the correlation function of the 4-th and 2-nd order models.

dependent sub-domains, until at the fifth scale, all the values $z(0), z(1), \ldots, z(127)$ have been generated, and thus, thanks to (3.43), all the values $y(0), y(1), \ldots, y(127)$ have been generated.

Evaluation of the multiscale models

Motivated by our preliminary analysis, we build four multiscale models, with the state dimension constrained to be no greater than 2, 4, 6 and 8, respectively.

In Figure 3-3, we display the correlation structure of the finest-scale of these processes. To carefully describe the content of these plots, we let ξ_0^λ denote the random vector comprising the finest-scale of the particular multiscale process in which the state vectors are constrained to have dimension no greater than λ; we thus have ξ_0^2, ξ_0^4, ξ_0^6 and ξ_0^8. We denote the i-th component of ξ_0^λ by $\xi_0^\lambda(i)$ (for $i = 0, 1, \ldots, 127$). Finally, to account for the fact that our multiscale models may lead to correlation structures that are only approximately stationary, we define

$$R_\lambda(n) \equiv \frac{1}{128-n} \sum_{j=0}^{127-n} E\left[\xi_0^\lambda(j+n)\xi_0^\lambda(j)\right], \quad n = 0, 1, \ldots, 127.$$

Figure 3-3 displays plots of $R_2(\cdot)$, $R_4(\cdot)$, $R_6(\cdot)$, and $R_8(\cdot)$, where this last function happens to coincide exactly with the desired correlation function. Consistent with our preliminary analysis, the 8-th order multiscale model is exact. The 6-th order model is very close to being exact, while there is noticeable degradation in the 4-th and 2-nd order models.

An alternative, and arguably more useful, way to measure model fidelity is in terms

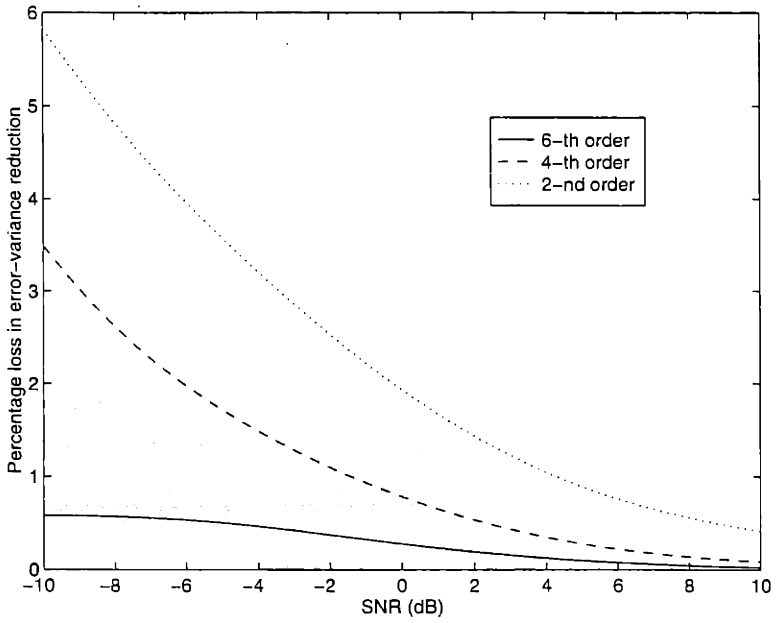


Figure 3-4: The percentage loss in error-variance reduction of our three reduced-order multiscale models.

of model performance in a designated application. Since one of the great strengths of our multiscale framework is the efficiency with which least-squares estimation can be performed, we treat least-squares estimation as the intended application. We focus specifically on the problem of estimating the value of the random vector χ_0,

$$\chi_0 \equiv \left(y(0) \quad y(1) \quad \cdots \quad y(127) \right)^T,$$

given the noisy observation $\chi_0+\nu_0$, where ν_0 is a zero-mean random vector having covariance equal to a multiple of the identity (i.e., equal to rI, for some scalar multiplying factor r). For any fixed value of r, we denote by $\hat{\chi}_0^{opt}$ the optimal linear least-squares estimate, and by $\hat{\chi}_0^{\lambda}$ the estimate associated with our multiscale model of order λ.

Our metric for model fidelity is related to loss in error-variance reduction that results from using $\hat{\chi}_0^{\lambda}$ rather than $\hat{\chi}_0^{opt}$. To define this metric precisely, we denote by p_i the prior variance of the process to be estimated, and also we denote by p_{opt} and p_λ the error variances of an optimal and a sub-optimal estimates, respectively. In our case, $p_i = R_{yy}(0) = 1$, while

$$p_{opt} = \frac{1}{128} E\left[\left(\chi_0 - \hat{\chi}_0^{opt}\right)^T \left(\chi_0 - \hat{\chi}_0^{opt}\right) \right], \qquad p_\lambda = \frac{1}{128} E\left[\left(\chi_0 - \hat{\chi}_0^{\lambda}\right)^T \left(\chi_0 - \hat{\chi}_0^{\lambda}\right) \right].$$

Since the optimal and sub-optimal estimates yield error variance reductions of $p_i - p_{opt}$ and $p_i - p_\lambda$, respectively, we associate with the sub-optimal estimate the following measure of fractional loss in error variance reduction (for a noise variance of r):

$$\Delta(r,\lambda) \equiv 1 - \frac{p_i - p_\lambda}{p_i - p_{opt}} = \frac{p_\lambda - p_{opt}}{p_i - p_{opt}}. \tag{3.45}$$

In Figure 3-4, we plot the values of $\Delta(r,\lambda)$, as r varies, for the values $\lambda = 2, 4$ and 6; the curve for $\lambda = 8$ is not included, because $\Delta(r,8) \equiv 0\ \forall r$. The abscissa of these plots has

k =	-2	-1	0	1	2
2		-0.0302	0.0592	-0.0407	
1	0.0406	-0.0980	0.2182	-0.0006	-0.0001
l = 0	-0.0836	0.4341		0.4341	-0.0836
-1	-0.0001	-0.0006	0.2182	-0.0980	0.0406
-2		-0.0407	0.0592	-0.0302	

Table 3.1: Autoregressive weights $\{r_{k,l}\}$ for the wool-texture WSMRF [39].

a logarithmic scale, where the signal-to-noise ratio (SNR), in dB, is related to r as

$$\text{SNR} \;=\; 10\log_{10}\frac{p_i}{r} \;=\; 10\log_{10}\frac{1}{r}.$$

By this criterion, even the 2-nd order model has quite good performance, with loss in error-variance reduction no greater than six percent for the range of SNRs displayed.

3.6.2 Reduced-order Representations of WSMRFs

We now turn our attention to a stationary WSMRF, having a fourth-order neighborhood structure and an autoregressive representation (2.8) that uses the weights given in Table 3.1. We define the field on a toroidal lattice, so that exact calculations, based on FFT techniques, are computationally feasible. We can then evaluate the loss associated with using reduced-order multiscale models as opposed to using a full-order multiscale model (or equivalently, as opposed to using statistically optimal FFT-based techniques). Figures 3-7a and 3-8a display mesh and contour plots respectively of the field's correlation structure, while Figure 3-6a displays a sample path of a field of dimension 256×256, drawn from the exact distribution using Gaussian deviates. This so-called wool texture is borrowed from [39].

We consider the problem of building a multiscale model, indexed on a quadtree, to realize this field on a 256×256 toroid. We constrain the model dimension, and then apply Algorithms 5 and 7, with the parameter θ_{rows} set to 516.

Preliminary Analysis

As in our first example, we begin with some preliminary analysis to determine a reasonable range of values to use for our model-dimension constraint λ_m. In keeping with our discussion in Sections 2.3 and 3.3.2, an exact realization will require complete retention of boundary information in the state vectors $x(s)$. To be more specific, let us define the random vector μ_s to contain boundary information of χ_s, exactly as in Section 3.4.3. In order to fully decorrelate χ_s from χ_{s^c} in the sense that

$$E\left[(\chi_s - E(\chi_s|\mu_s))(\chi_{s^c} - E(\chi_{s^c}|\mu_s))^T\right] \;=\; \mathbf{0},$$

we must choose μ_s to contain the values of χ_s over a boundary of width 2; this fact follows directly from the fourth-order Markov structure of the wool texture. Straightforward calculation then reveals that for χ_s representing the values of the field over a $K \times K$ block of pixels with $K \geq 4$, the needed boundary information requires that μ_s have dimension $8(K-2)$. Thus, an exact multiscale realization requires a scale-varying state dimension. For a lattice of dimension 256×256, the state at the root node will have dimension $4 \cdot 8(128-2) = 4032$,

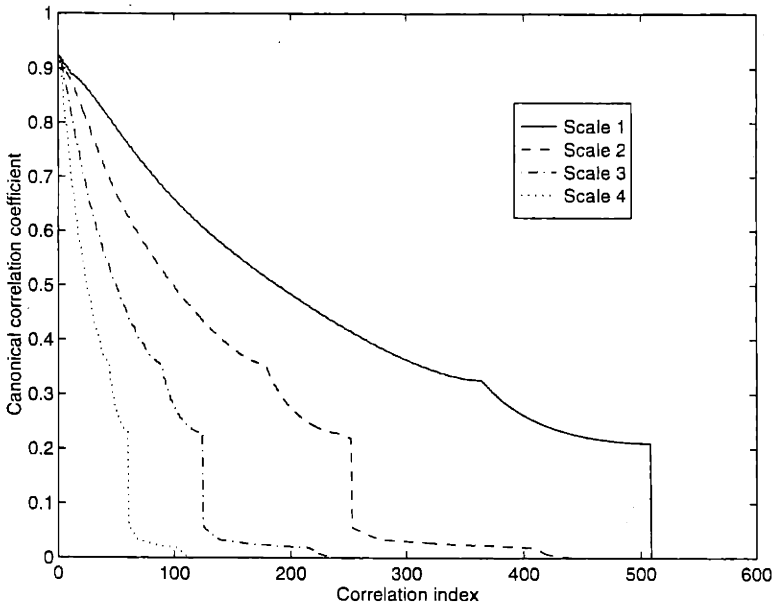


Figure 3-5: Plots of the values of canonical correlation coefficients between μ_s and μ_{s^c} for $m(s) = 1$, 2, 3 and 4. In each case, both μ_s and μ_{s^c} have dimension equal to $\theta_{rows} = 516$.

each of the four states at the next finest scale will have dimension $4 \cdot 8(64 - 2) = 1984$, and so forth. In contrast to our preceding 1-D example, we will here have to reduce drastically the state dimension, in order to obtain models of practical use.

We can obtain additional insight by examining the values of the canonical correlation coefficients between μ_s and μ_{s^c}. The random vectors μ_s and μ_{s^c} are defined in step 3 of our modeling algorithm (i.e., Algorithm 7), while the canonical correlation coefficients between μ_s and μ_{s^c} are calculated in step 4. In Figure 3-5, we plot the values of these coefficients, for nodes s at scales $m(s) = 1$, 2, 3 and 4. In the context of building reduced-order models, the displayed values do not appear encouraging; for all four displayed scales, there are many large-valued canonical correlation coefficients. For instance, there are 64, 34, 17 and 8 coefficients at the respective scales 1, 2, 3 and 4 that exceed the value 0.75. However, as we demonstrate next, the plot in Figure 3-5 is quite misleading; we can actually do quite well with a reduced-order model of far lower dimension than suggested by Figure 3-5, if we measure model fidelity with respect to either of the criteria used in the previous example (i.e., the match between desired and realized correlation, or alternatively the fractional loss in error-variance reduction, in linear least-squares estimation.)

Evaluation of the multiscale models

Let us evaluate the performance of three multiscale models, in which the state dimension is constrained to be no greater than 16, 8 and 4, respectively. In Figures 3-6b, c and d, we display sample paths generated using our three multiscale models. A visual comparison of these sample paths with the one in part a suggests that all of our multiscale models, even the 4-th order one, have captured the important qualitative statistical characteristics of the wool texture. A careful inspection of (b), (c) and (d) will reveal slight discontinuities at the quadrantal boundaries (e.g., at the horizontal and vertical mid-lines), but these effects diminish as model order increases and are barely discernible in the 16-th order model.

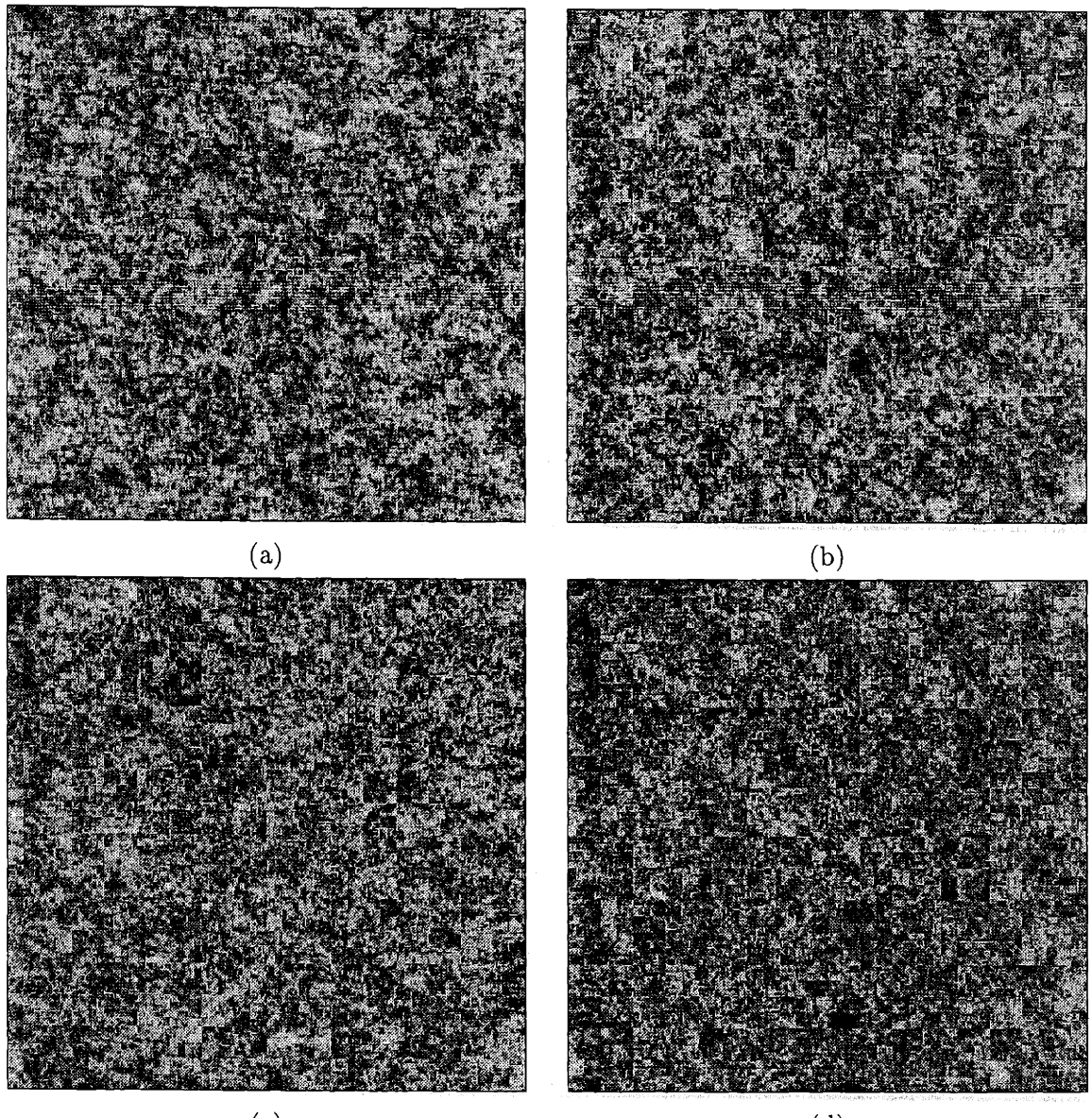


Figure 3-6: These four figures display sample paths of the wool-texture WSMRF, for a 256×256 pixel region. The sample path in (a) is drawn from the exact distribution, using FFT techniques. The paths in (b), (c) and (d) are drawn from distributions that approximate the exact distribution; they come multiscale models in which the state dimension is constrained to be less than or equal to 16, 8 and 4, respectively.

In Figures 3-7b, c, and d and 3-8b, c, and d, we display mesh and contour plots, respectively, of the correlation structure of the finest-scale of these multiscale processes. As in the previous example, we must proceed with caution here, since our reduced-order models lead to correlation structures that are only approximately stationary. We let ξ_0^λ denote the random vector comprising the finest-scale of the particular multiscale process in which the state vectors are constrained to have dimension no greater than λ; we thus have ξ_0^4, ξ_0^8, and ξ_0^{16}. We denote the (i,j)-th component of ξ_0^λ by $\xi_0^\lambda(i,j)$ (for $i,j = 0,1,\ldots,255$). In terms of these conventions, the plots in Figures 3-8b, c, and d display contours of the function $R_\lambda(\cdot,\cdot)$, for $\lambda = 16$, 8 and 4 respectively, where

$$R_\lambda(m,n) \equiv \frac{1}{256^2}\sum_{i=0}^{255}\sum_{j=0}^{255} E\left[\xi_0^\lambda\left((i+m) \bmod 256, (j+n) \bmod 256\right)\xi_0^\lambda(i,j)\right].$$

To facilitate comparison of these plots, Figures 3-9a and b display horizontal and vertical slices of the contours, where in each case, all of the slices are overlayed on a single plot. These figures make clear the steady improvement in fidelity that is associated with increasing the model order. We remark that the correlation function for each of our multiscale models has been evaluated by Monte Carlo simulation. In particular, we have generated N_{samp} sample paths of the field, and have then averaged together the sample correlation functions of these sample paths. Using the insights gathered in Section 2.2.3, and in particular (2.23), we have chosen the value N_{samp} to be sufficiently large so that with 95 percent confidence, each estimated value of $R_\lambda(m,n)$ is with 0.005 of its actual value. Since a variation of 0.005 is roughly on the order of the width of the plotted lines, no error bars are needed here.

Finally, just as in our previous example, we evaluate the fidelity of these multiscale models in terms of their performance in a linear least-squares estimation context. Letting χ_0 be a random vector containing the values of the wool-texture random field over a 256×256 region, we consider the problem of estimating χ_0, given the noisy observation $\chi_0+\nu_0$, where ν_0 is a zero-mean random vector having covariance rI. For any fixed value of r, we denote by $\hat{\chi}_0^{opt}$ the optimal linear least-squares, and by $\hat{\chi}_0^\lambda$ the estimate associated with our multiscale model of order λ.

Our metric for model fidelity is once again percentage loss in error-variance reduction. In Figure 3-10, we display this loss for a number of different SNRs. The values in the plot have been calculated by Monte-Carlo simulation; using the insights gathered in Section 2.2.3, and in particular (2.20), we have chosen the number trials to be great enough to ensure that the variance loss percentages are within 0.5% of their true value with 95 percent confidence. By this criterion, even this 4-th order model does a very respectable job, yielding estimation results that are within roughly 5% of the optimal estimator.

3.6.3 Reduced-order Representations of Isotropic Random Fields

We now turn away from WSMRFs, to build multiscale representations for a random field that is stationary, but is explicitly *not* a WSMRF. The particular correlation function we examine is an isotropic one that is of considerable interest in the geological sciences [59]; the function can be expressed analytically as follows:

$$R_{yy}(k,l) = R_{yy}(r) = \begin{cases} (1-3/2(r/\ell)+1/2(r/\ell)^3) & 0\le r\le \ell, \\ 0 & r>\ell, \end{cases} \tag{3.46}$$

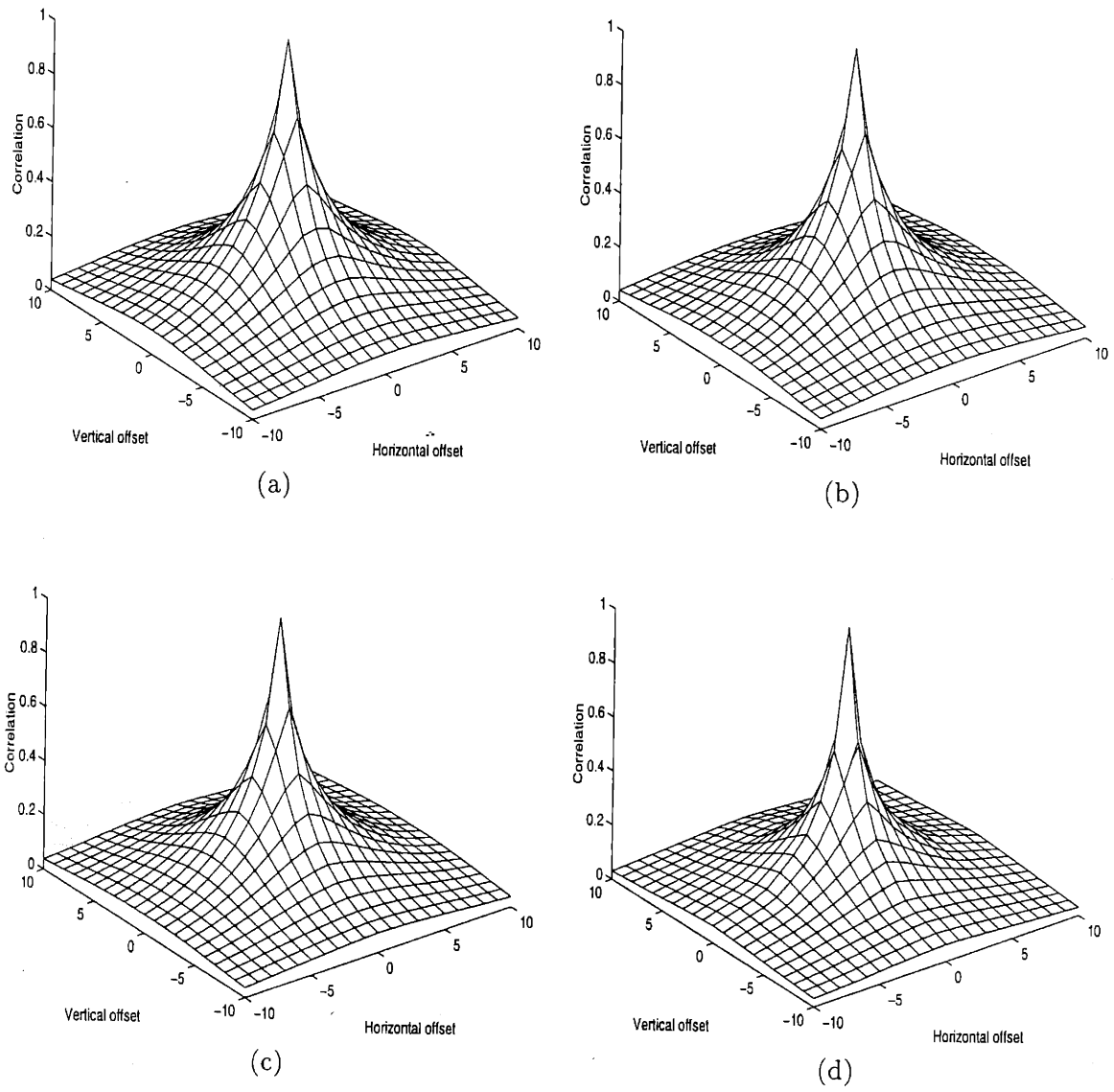


Figure 3-7: These four figures display mesh plots of the correlation function of the wool texture. The mesh plot in (a) represents the exact correlation function, as determined by FFT techniques. The contour plots in parts (b), (c) and (d) represent the correlation function of the finest scale of the 16-th, 8-th and 4-th order multiscale models, respectively. The contour plots in parts (b), (c) and (d) have been determined by Monte-Carlo simulation, using enough trials so that with 95 percent confidence, every estimated correlation value is within 0.005 of its correct value. Since a variation of 0.005 is roughly the order of the width of the plotted lines, no error bars are needed.

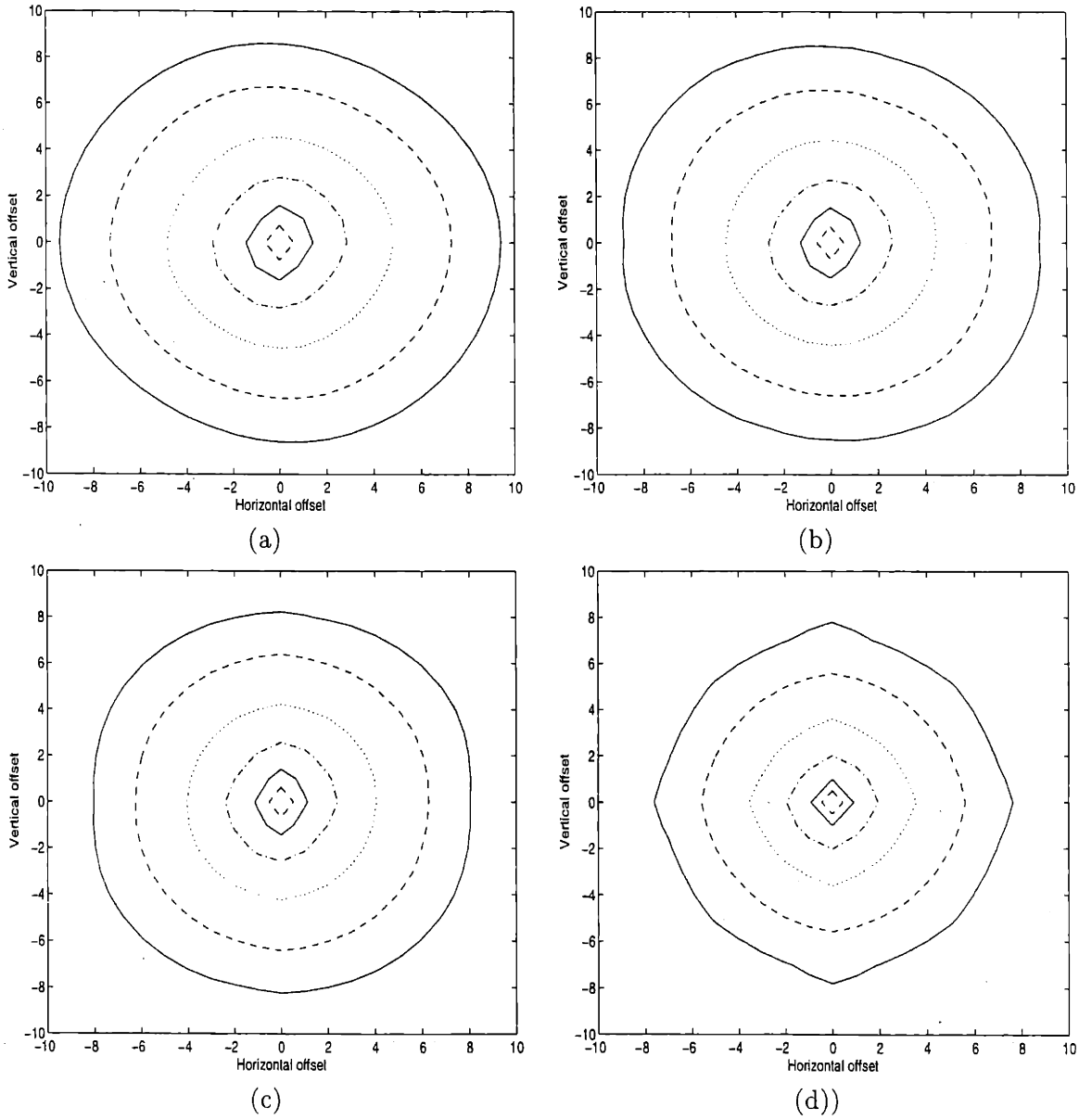


Figure 3-8: These four figures display contour plots of the correlation function of the wool texture, with the contour levels at 0.8, 0.6, 0.4, 0.25, 0.15, and 0.1. The contour plot in (a) represents the exact correlation function, as determined by FFT techniques. The contour plots in parts (b), (c) and (d) represent the correlation function of the finest scale of the 16-th, 8-th and 4-th order multiscale models, respectively. The contour plots in parts (b), (c) and (d) have been determined by Monte-Carlo simulation, using enough trials so that with 95 percent confidence, every estimated correlation value is within 0.005 of its correct value.

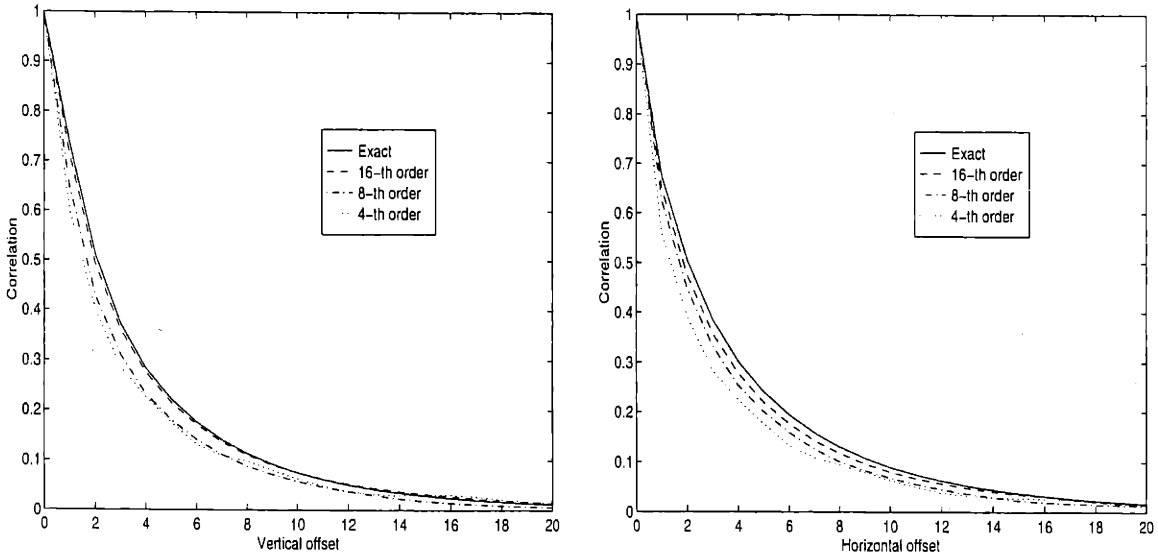


Figure 3-9: Comparison of (a) vertical and (b) horizontal slices of the correlation contour plots in the previous figure. Again, these plots are based on Monte-Carlo simulation, where each point is within 0.005 of its correct value with 95 percent confidence.

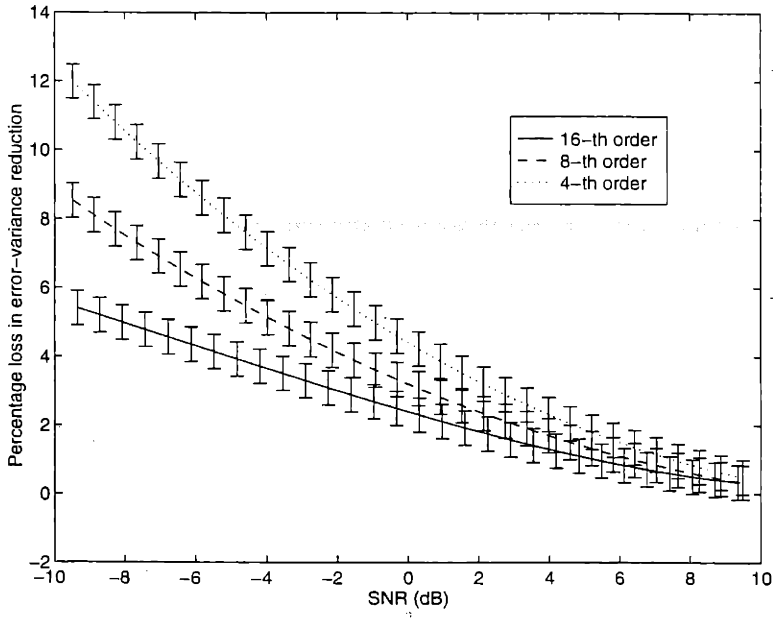


Figure 3-10: The percentage loss in error-variance reduction of our three reduced-order multiscale models. We have used Monte-Carlo simulation to calculate these percentage losses, using enough trials to ensure that the calculated percentages are with 0.5% of their true value, with 95 percent confidence; the displayed error bars reflect this confidence.

where $r = \sqrt{k^2 + l^2}$, and ℓ is the characteristic length of the function. A plot of this function for $\ell = 80$ is represented by the solid curve in Figure 3-12; we see from this plot that there is significant long-range correlation, at least relative to the total grid size (i.e., 128×128) we will be using.

We consider the problem of building a multiscale model, indexed on a quadtree, to realize the correlation function (3.46) on a 128×128 grid. We constrain the multiscale model dimension to the respective values of 64, 32, 16 and 8, and then apply Algorithms 5 and 7, with the parameter $\theta_{rows} = 260$.

In Figure 3-11a, we display as a contour plot the exact correlation function (3.46). Then, in Figures 3-11b, c and d, we display as contour plots the correlation function associated with our multiscale models of order 32, 16, and 8, respectively. We do not include a contour plot for our model of order 64, because for orders greater than just 16, our multiscale models capture virtually all of the significant correlation structure. This fact is reinforced in Figures 3-12a and b, where we display horizontal and vertical slices of these contour plots.

In Figure 3-13, we display sample paths of this random field, generated with our models of order 64, 32, 16 and 8, and using Gaussian deviates. In contrast to the foregoing criterion, for which the 16-th order model did an excellent job, we here see that such a low-order model leads to visually distracting blocky artifacts at the quadrantal boundaries. While in many applications, these artifacts are devoid of any statistical significance, they may be important in other contexts. One way to eliminate these artifacts is employ a relatively high-order model multiscale model; for instance, as shown in Figure 3-13a, the 64-th order model is effective in this regard. An alternative, arguably more elegant approach to eliminating these artifacts is pursued in detail in Chapter 5.

Finally, let us consider the use of these multiscale models to carry out linear least-squares estimation. In Figure 3-14a, we display the original signal that we will be attempting to estimate. This signal consists of 128×128 pixels and has a Gaussian distribution. It is drawn from the *exact* distribution implied by (3.46) with $\ell = 80$. This is effected by embedding the 128×128 grid into a larger 256×256 toroidal lattice, and extending the definition of $R_{yy}(\cdot, \cdot)$ to have periodic boundary conditions; for $\ell = 80$, this approach leads to a valid (i.e., positive definite) correlation function.

We consider two estimation problems related to the signal in Figure 3-14a. For the first, we corrupt the signal with spatially stationary white noise having covariance one, thus leading to an SNR of 0dB (since the signal also has a variance of one, as indicated by (3.46)). In Figure 3-14b, we display an estimate based on our multiscale model of order 64. The MSE here is 0.0498. While there is no computationally feasible way to determine the mean-square error of an optimal estimator for this problem, we can obtain a fairly tight lower bound for the optimal MSE. In particular, let us consider the problem of estimating the value of the 256×256 signal, from which our 128×128 signal has been extracted. Since this 256×256 signal is stationary and is indexed on a toroidal lattice, exact calculations are possible. In particular, for estimating this signal in 0dB white noise, the optimal, FFT-based estimator has an MSE of 0.0458, which must lower-bound the MSE of an optimal estimator in our original estimation problem. By comparison, then, our measured MSE of 0.0498 is quite satisfactory. Although not shown in the Figure, the same level of performance is also achieved by our lower-order multiscale models; specifically, our models of order 32, 16 and 8 achieve MSEs of 0.0501, 0.0533 and 0.0544, respectively, which are all close to the optimal.

We now turn our attention to an estimation problem for which the FFT is of no practical use at all. In particular, we now consider the problem of estimating the signal displayed

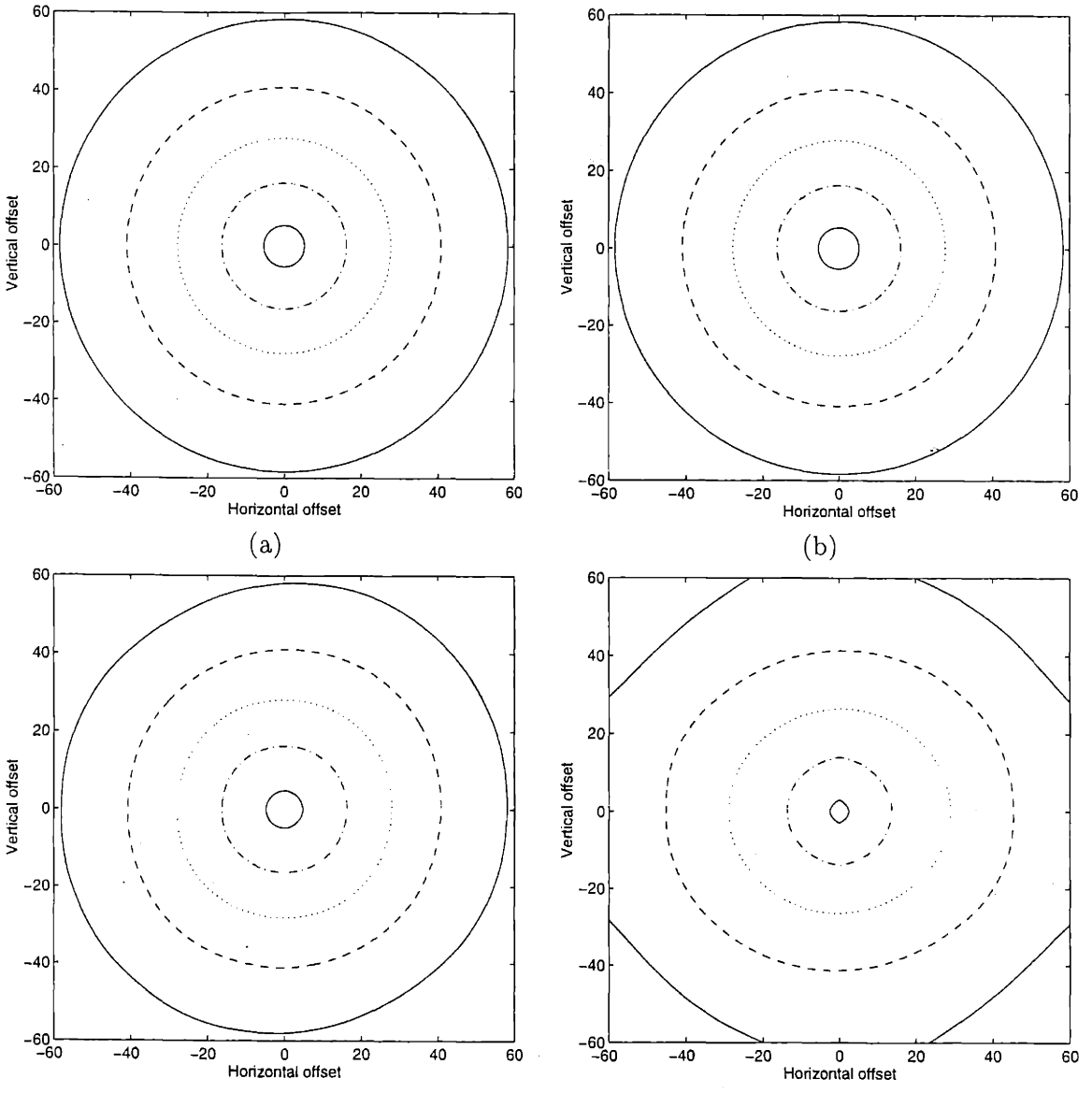


Figure 3-11: These four figures display contour plots associated with $R_{yy}(\cdot,\cdot)$, defined in (3.46), with the contour levels at 0.9, 0.7, 0.5, 0.3 and 0.1. (a) The exact, desired correlation function. (b), (c), and (d) The correlation function associated with multiscale models of order 32, 16 and 8, respectively. These three have been determined by Monte-Carlo simulation, using enough trials so that every estimated correlation value is within 0.005 of its correct value.

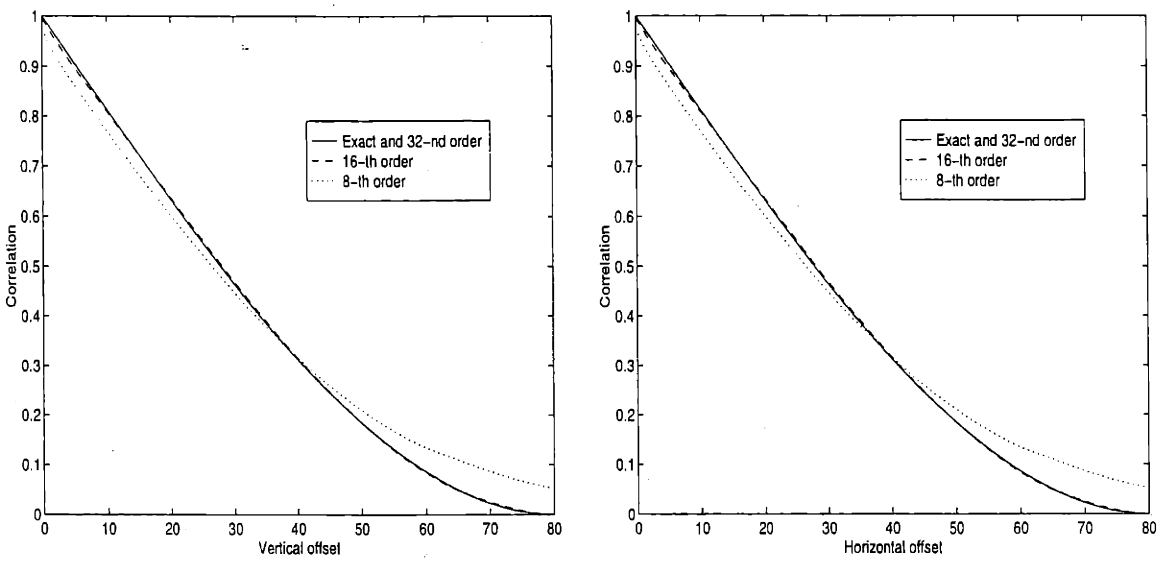


Figure 3-12: Comparison of (a) vertical and (b) horizontal slices of the correlation contour plots in the previous figure. Again, these plots are based on Monte-Carlo simulation, where each point is within 0.005 of its correct value with 95 percent confidence.

in Figure 3-14a, based on noiseless measurements at the extremely sparse set of points displayed in Figure 3-14c. These points provide only 1.11% coverage of the image region. Their irregular distribution is the key reason that FFT techniques are not applicable. On the other hand, in Figure 3-14d, we display the estimate that results from use of our multiscale model of order 64. In light of the sparsity of our measurement coverage, this estimate has impressively captured the coarse qualitative features of the true signal; in fact, the MSE of this estimate is only 0.1147, i.e., about 90% variance reduction.

3.6.4 Reduced-order Representations of Warped-version of Isotropic Correlation Function

For our last example, we build multiscale representations for a stationary random field having a correlation function that is a warped version of the isotropic correlation function $R_{yy}(k,l)$ in (3.46. Our warped version, which we denote by $R'_{yy}(k,l)$ is defined as follows:

$$\begin{aligned} R'_{yy}(k,l) &= R_{yy}(k',l'), \\ \begin{pmatrix} k' \\ l' \end{pmatrix} &= \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} k \\ l \end{pmatrix}, \\ \theta &= \frac{\pi}{4} - \frac{\pi}{13}. \end{aligned} \tag{3.47}$$

The characteristic length ℓ of $R_{yy}(k,l)$ (see (3.46) is again set to $\ell = 80$. A contour plot of $R'_{yy}(k,l)$ is displayed in Figure 3-15a, while slices of this correlation function along the directions of strongest and weakest correlation are displayed in Figures 3-16a and b, respectively.

We consider the problem of building a multiscale model, indexed on a quadtree, to realize the correlation function (3.47) on a 128×128 grid. We constrain the multiscale

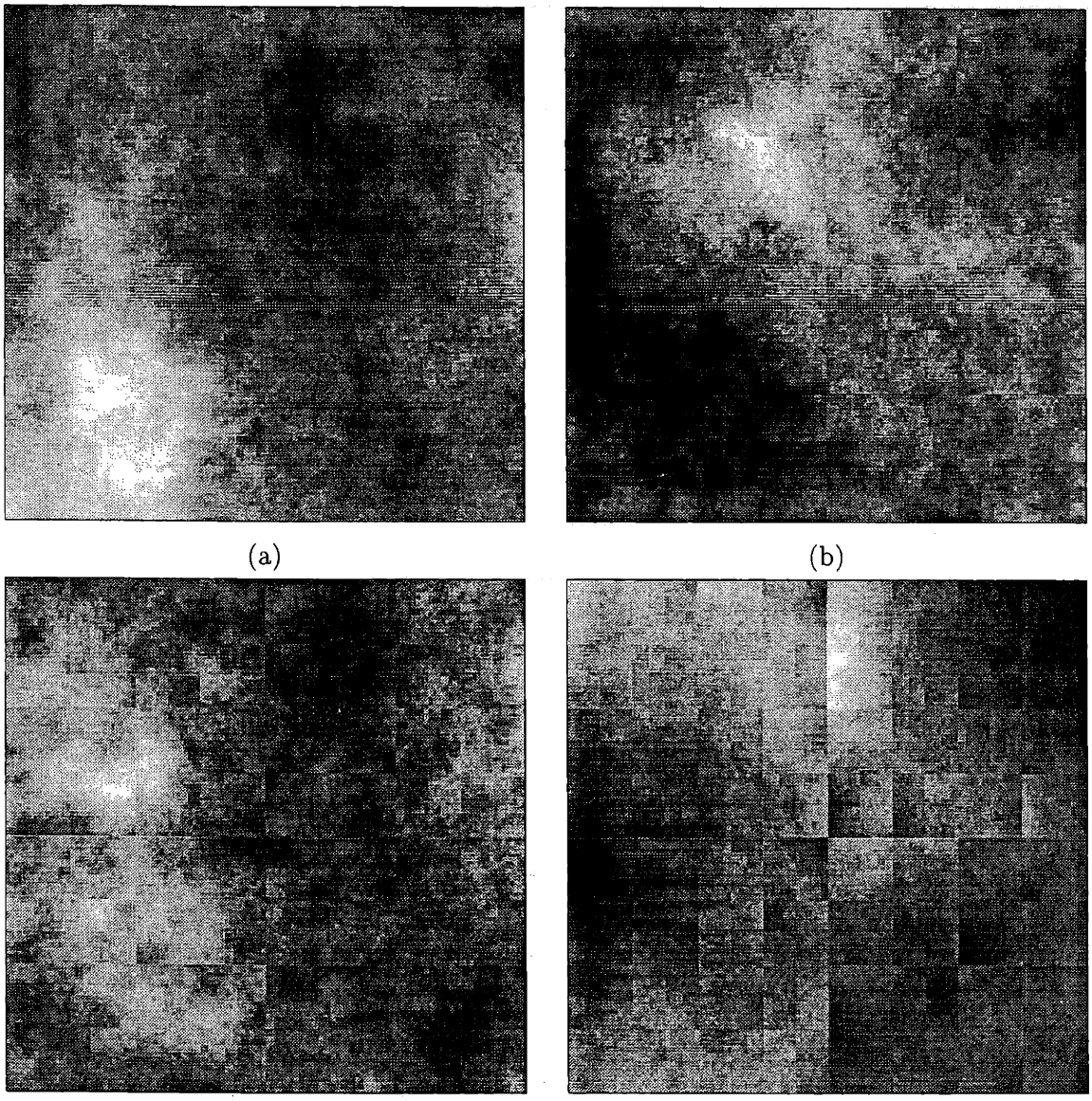


Figure 3-13: These four figures display sample paths of random fields having approximately the isotropic correlation function given in (3.46). The sample paths each have 128×128 pixels, with (a), (b), (c) and (d) corresponding, respectively, to multiscale models of order 64, 32, 16 and 8, using Gaussian deviates. We see that a relatively high-order model is required to eliminate the blocky artifacts at the quadrantal boundaries.

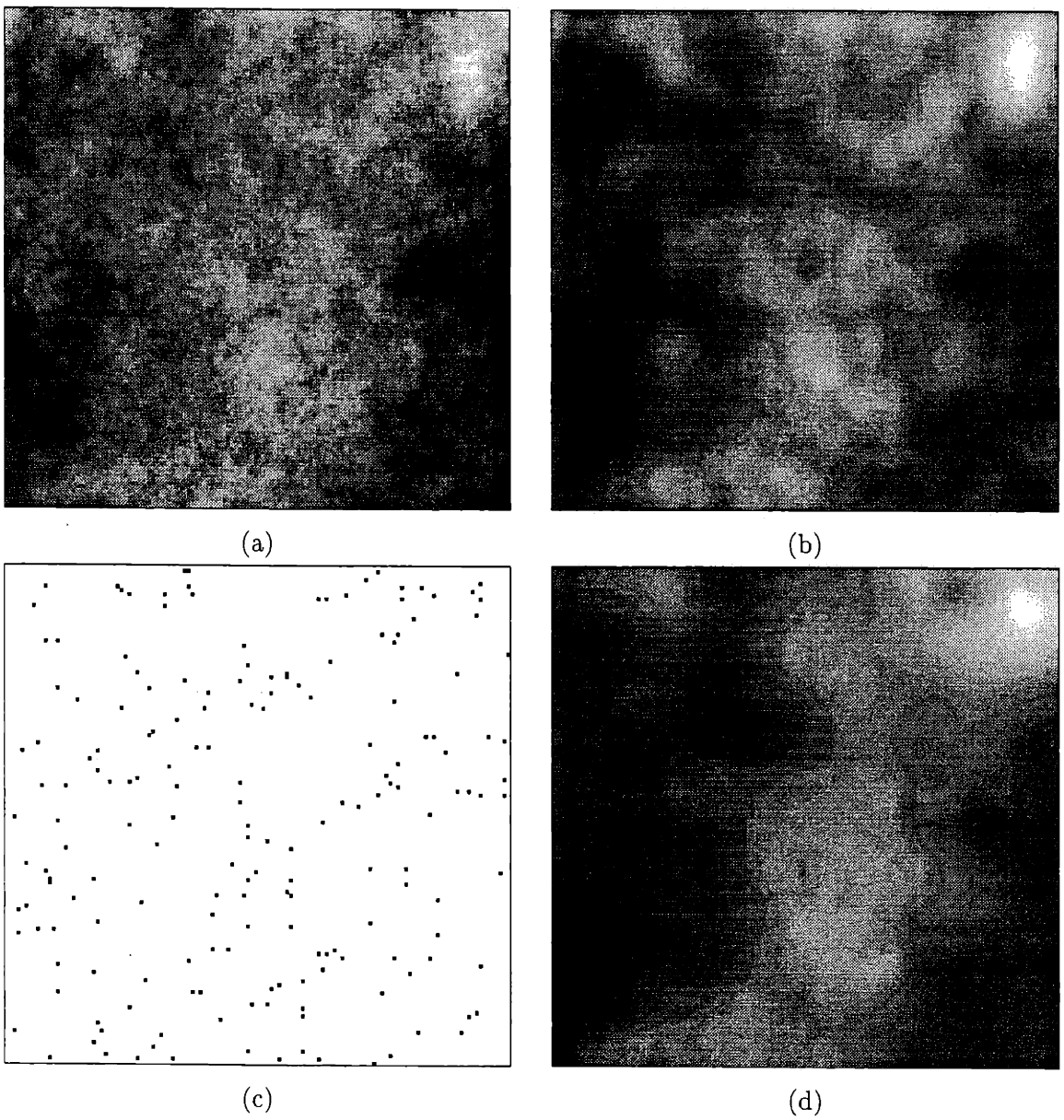


Figure 3-14: These four figures relate to linear least-squares estimation of a signal having the isotropic correlation function in (3.46). (a) The original signal, with Gaussian deviates, drawn from the exact distribution using FFT-based techniques. (b) Estimate of the sample path in (a), based on noisy, densely distributed measurements of the signal, with 0dB SNR; a 64-th order multiscale model is used to obtain this estimate. (c) Locations of observed pixels, for a second estimation experiment; these observed pixels provide only 1.11 % coverage of the image. (d) Estimate of the sample path in (a), based on noiseless observations of the observed pixels (displayed in (c)).

model dimension to the respective values of 64, 32, 16 and 8, and then apply Algorithms 5 and 7, with the parameter $\theta_{rows} = 260$.

In Figures 3-15b, c and d, we display as contour plots the correlation function associated with our multiscale models of order 32, 16, and 8, respectively. We do not include a contour plot for our model of order 64, because at this order, the contour plot is indistinguishable from the ideal, desired correlation in (a). To allow for more direct comparison of these contours, we overlay slices of them in Figures 3-16a and b; more specifically, Figure 3-16a represents a slice of the contour plots, along the direction of strongest correlation, while part b represents a slice of the contour plots along the direction of weakest correlation.

In Figure 3-17, we display sample paths of this random field using Gaussian deviates, generated with our models of order 64, 32, 16 and 8. We see that unless a relatively high order model is used, the sample paths exhibit visually distracting blocky artifacts at the quadrantal boundaries. Again, we emphasize that in Chapter 5, we will describe an elegant way to eliminate these artifacts.

3.7 Parameterization by W_s Matrices: A Closer Look

In essence, our approach to multiscale modeling can be summarized as follows: (i) we start with a q-th order tree and a desired, finest-scale covariance P_{x0}; (ii) we determine, for each node s in the tree, a matrix W_s that parameterizes the information content of the state vector $x(s)$; (iii) we calculate, from these W_s matrices, values for the parameters $P(0)$, $A(s)$ and $B(s)$, via (3.5), (3.9) and (3.10), respectively. Thus, any model produced in this fashion can be characterized completely by the set $\{P_{x0},\ W_s\}_s$, from which the parameters $\{P(0),\ A(s), B(s)\}_s$ follow uniquely.

While the numerical experiments in the previous section clearly demonstrate the practical viability of this modeling approach, there remain unaddressed certain theoretical issues regarding our parameterization by $\{P_{x0},\ W_s\}_s$. We note, for instance, that while we can always map from $\{P_{x0}, W_s\}_s$ to $\{P(0), A(s), B(s)\}_s$, we cannot always go in the reverse direction, even with exact realizations; in other words, for a given covariance P_{x0} and parameters $\{P(0),\ A(s), B(s)\}_s$ that realize this covariance exactly, *there may not exist a corresponding set* $\{W_s\}_s$. A similar phenomenon occurs in the time-series realization context, where the consequences are well understood, and by comparing the two cases, we will obtain some interesting insights.

The other issue we explore is the effect of our $\{W_s\}_s$ parameterization on the interscale propagation of state information. This propagation is handled only implicitly by our myopic modeling approach, and more specifically by our use of the $\{W_s\}_s$ parameterization. This observation leads us to develop an alternative modeling approach that is less myopic, thereby allowing the propagation of information to be handled with tighter control. The difficulty with this alternative is the demanding nature of the required bookkeeping. In fact, this bookkeeping prohibitively stresses both memory and computational resources in problems of practical size and interest. Thus, by way of contrast, we ultimately underscore the practical soundness of our original modeling approach.

3.7.1 Internal Vs. External Realizations

In Section 3.2, we saw in the time-series context that under fairly general conditions, an exact, minimal realization of a vector-valued random process $y(n)$, evolving in discrete time, can be achieved with a state-space model in which the process state $z(n)$ is a linear

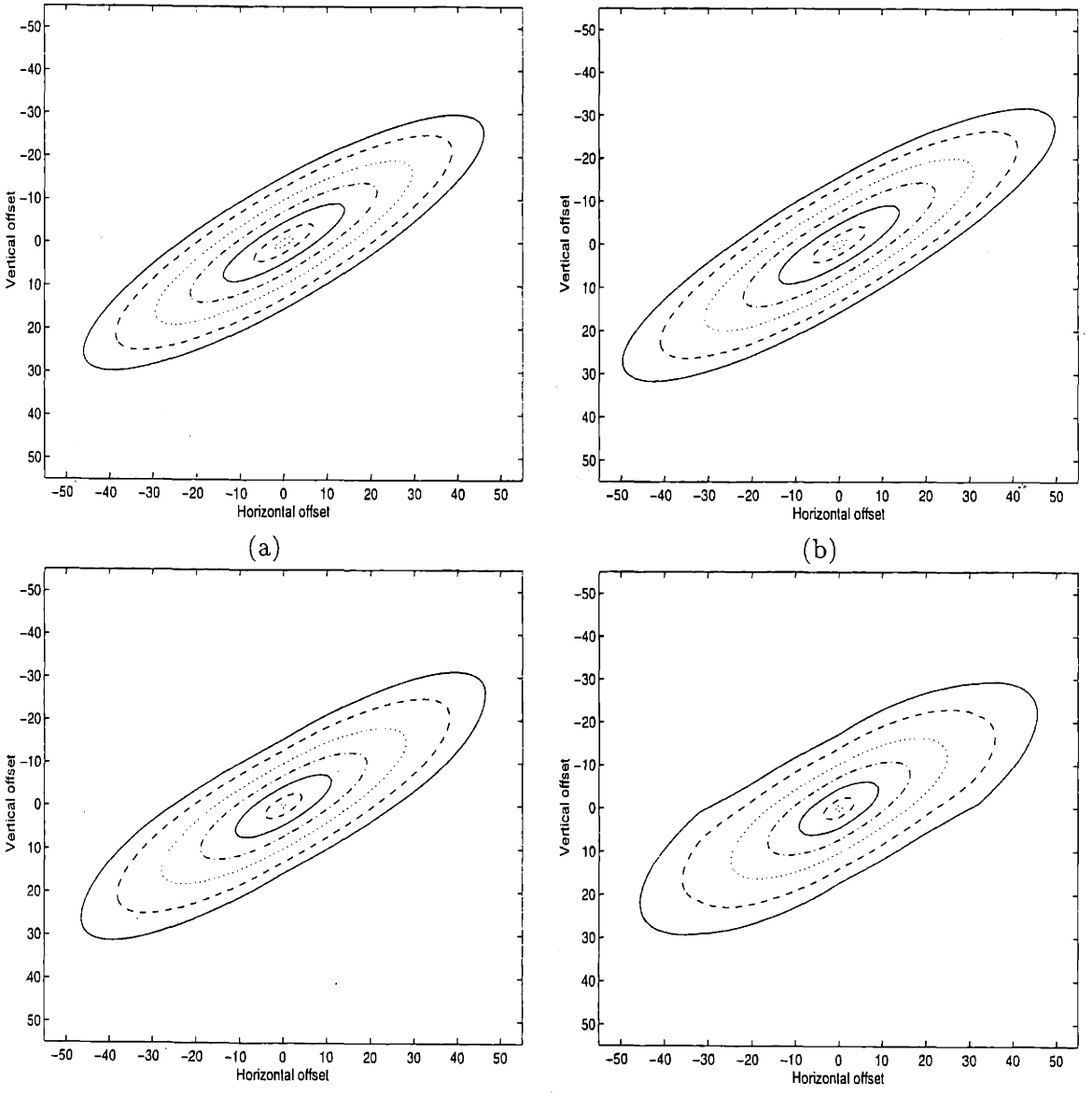


Figure 3-15: These four figures display contour plots associated with $R'_{yy}(\cdot,\cdot)$, defined in (3.47), with the contour levels at 0.95, 0.85, 0.75, 0.6, 0.45, 0.3 and 0.15. (a) The exact, desired correlation function. (b), (c), and (d) The correlation function associated with multiscale models of order 32, 16 and 8, respectively. These three have been determined by Monte-Carlo simulation, using enough trials so that every estimated correlation value is within 0.005 of its correct value.

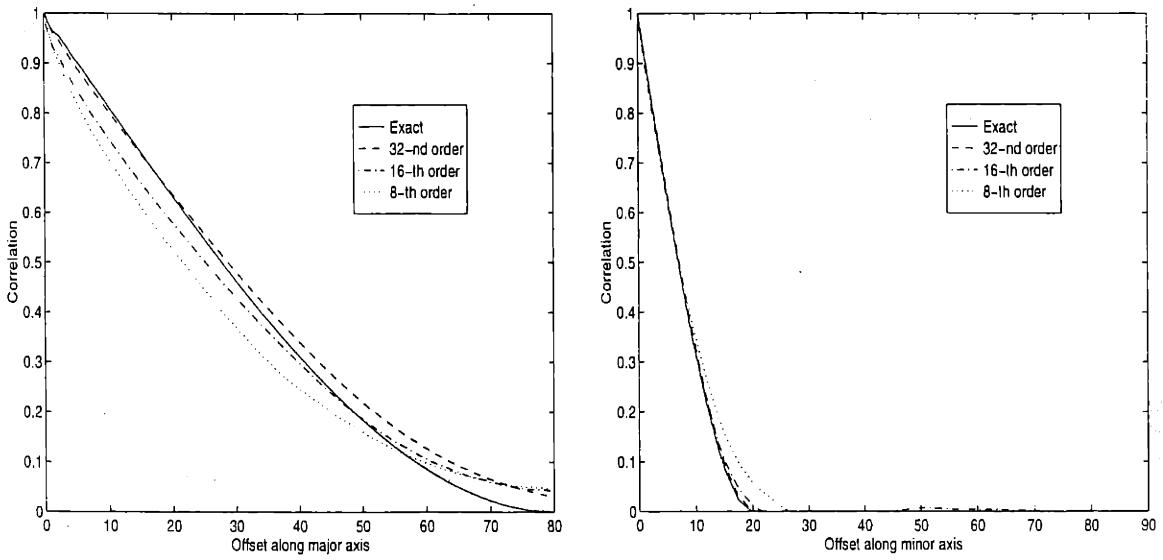


Figure 3-16: Comparison of slices of correlation contour plots in the previous figure. (a) A slice along the direction of the major axis of the ellipses in part (a) of the previous figure. (b) A slice along the direction of the minor axis of the ellipses in part (a) of the previous figure. Again, these plots are based on Monte-Carlo simulation, where each point is within 0.005 of its correct value with 95 percent confidence.

function of either the past of $y(\cdot)$ or the future of $y(\cdot)$ (i.e., either $\eta_{past}(n)$ or $\eta_{future}(n)$ in (3.3)). This idea is formalized in [41], where such a realization is termed an *internal* one, because everything internal to the state-space model (i.e., $z(n)$ and $w(n)$) is obtainable directly from the observed process $y(\cdot)$. A standard example of an internal realization is the so-called *innovations* representation, in which the driving noise is the innovations process produced by either a forward-running or backwards-running Kalman filter associated with *any* state-space realization of the process [2,41,56]. Our main point here is simply that internal realizations are readily obtainable in the time-series context, and under fairly general conditions, they constitute a rich enough class of models to include minimal realizations.

It is natural to generalize the internal-realization concept to the multiscale context, where, for obvious reasons, we refer to multiscale models that are parameterized by $\{P_{x0}, W_s\}_s$ as *internal multiscale realizations*. Given the richness of internal realizations in the time-series context, let us consider their richness in the multiscale context. As a vehicle for our development, we consider the problem of building a multiscale model, indexed on a dyadic tree, to realize exactly the following finest-scale covariance:

$$P_{x0} = \begin{pmatrix} 3 & 2 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 1 & 1 & 3 & 2 \\ 1 & 1 & 2 & 3 \end{pmatrix}. \tag{3.48}$$

By direct calculation, one can verify that this covariance is realizable with a multiscale model consisting of three scales, in which all states have dimension of one, the covariance $P(0)$ has value one, and all transition matrices $A(s)$ and noise-shaping matrices $B(s)$ have values of one. Because a unity state dimension is the minimum possible, this suggested realization must be minimal. On the other hand, there does not exist a set $\{P_{x0}, W_s\}_s$

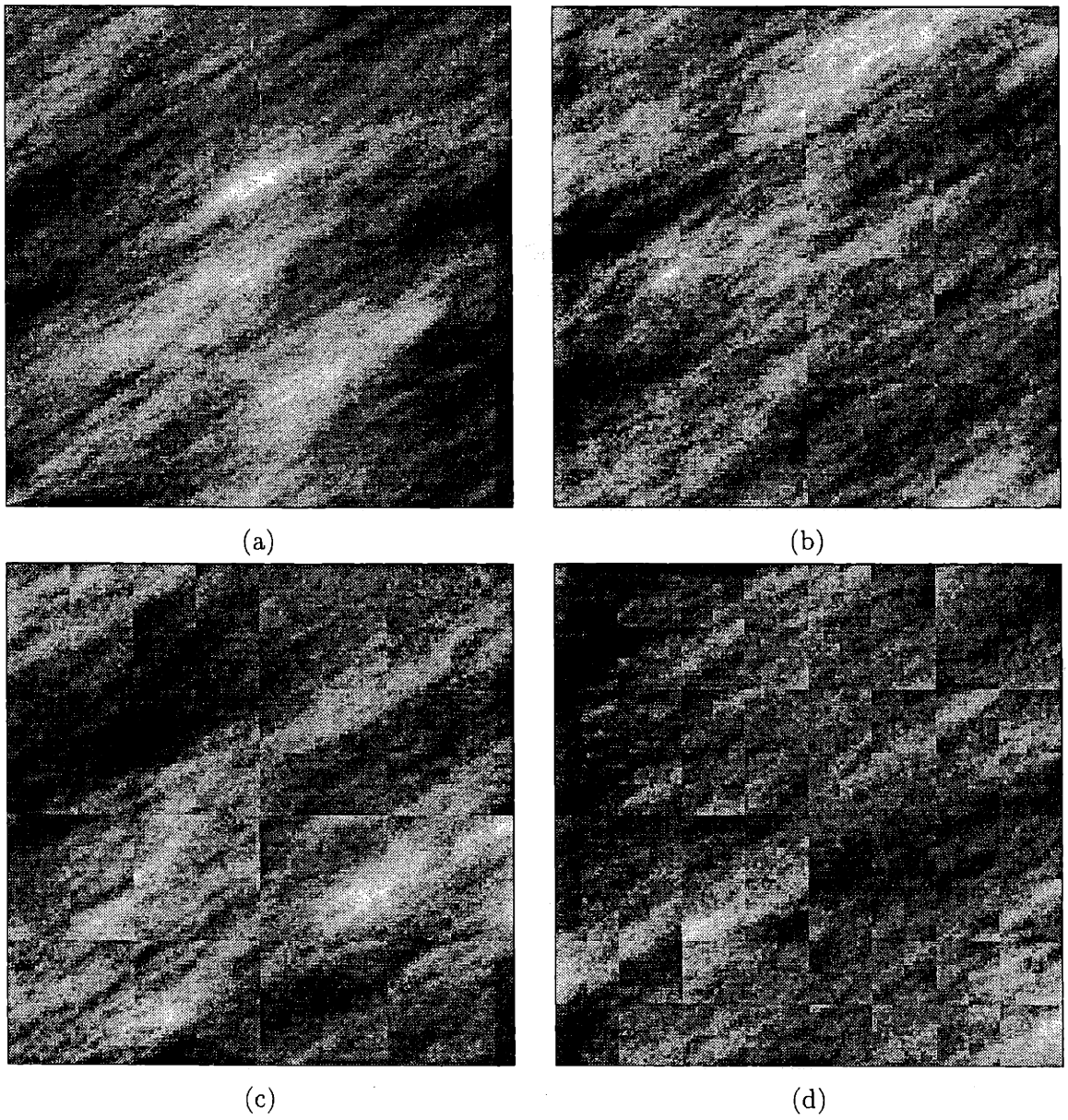


Figure 3-17: These four figures display sample paths of a random field having the correlation function given in (3.47, for a 128×128 pixel region. The sample paths in (a), (b), (c) and (d) correspond to multiscale models of order 64, 32, 16 and 8, respectively, using Gaussian deviates. Just as in our previous example, we see that a relatively high-order model is required to eliminate the blocky artifacts at the quadrantal boundaries.

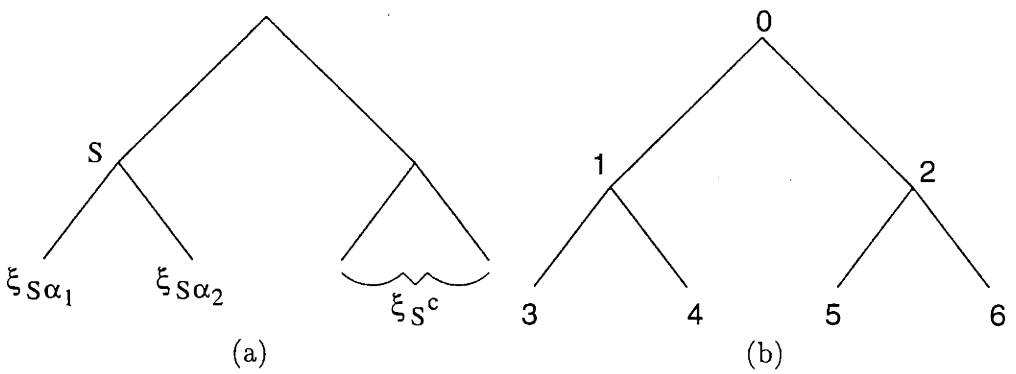


Figure 3-18: Notational conventions for our use of dyadic trees to realize (a) P_{x_0}, given in (3.48), and (b) P_{x_0}, given in (3.49).

that leads to a minimal, exact realization of (3.48); in other words, there is no internal realization of (3.48) having a state dimension of one at every node. The reason is that for the node labeled s in Figure 3-18a, any matrix W_s that exactly fulfills (3.8) will have at least two rows, thus leading to a state vector $x(s)$ having dimension at least two, which is not minimal. We demonstrate in Appendix C this fact regarding the matrix W_s.

Thus, we have uncovered an interesting and non-trivial difference between Gauss-Markov time-series processes and multiscale processes. While in the former case, the class of internal realizations is sufficiently rich to include minimal realizations, the same is not generally true in the multiscale context.

3.7.2 Propagation of Information from Scale to Scale

We now turn our attention to the issue of interscale propagation of information. To see the issues involved, let us consider the problem of building a multiscale model, indexed on a dyadic tree, to realize exactly the following finest-scale covariance:

$$P_{x_0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.49}$$

Just as with the covariance in (3.48), an exact realization will here require a dyadic tree having three scales. For convenience, we index the seven nodes in this tree by 0, 1, ..., 6, as illustrated in Figure 3-18b. One possible internal, exact realization uses the following values for the W_s matrices:

$$\begin{aligned} W_0 &= \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}, \\ W_1 &= \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad W_2 = \begin{pmatrix} 1 & 0 \end{pmatrix}, \\ W_3 &= W_4 = W_5 = W_6 = 1. \end{aligned} \tag{3.50}$$

A valid alternative, which also leads to an exact, internal realization, is to replace W_0 in (3.50) with W_0',

$$W_0' = \text{diag}(1,1,1,1),$$

while retaining the same values for $W_1, W_2, \ldots, W_6$ as in (3.50). There is an important difference between the models that result from these two choices for the W_s matrices. The first choice leads to a model in which coarse-scale information is preserved in its journey to the finest scale; in particular, we see that $x(s) = W_s\xi_s$. On the other hand, the second choice leads to a model in which information is lost in its journey to the finest scale. In fact, by using W_0' in lieu of W_0, we have somewhat perversely created a multiscale model in which the *entire* finest-scale process is generated at the root node, and then some of this information is immediately discarded in the transition to the middle scale, whence new values for this discarded information are generated in the transition to the third, finest scale. Although the finest scale process does have the correct, desired correlation, care must be exercised in interpreting the information content of the coarsest-scale state. In particular, our parameterization $x(0) = W_0\chi_0$ is misleading, in the sense that actually, $x(0) \neq W_0\xi_0$. The source of this problem is the implicit way that information is propagated from scale to scale.

3.8 Alternative Realization Approach: Explicit Handling of Information Propagation

We now develop an alternative realization approach that handles more explicitly the propagation of information from scale to scale. For simplicity, we limit our attention to multiscale processes indexed on the dyadic tree. Our objective is to build a multiscale model, indexed on the given tree, such that the covariance P_{ξ_0} of the resulting finest-scale process *exactly* matches P_{χ_0}. While one solution to this problem was described in Section 3.3.2, that solution was tied to the W_s parameterization, which we explicitly wish to avoid here. Furthermore, although we focus on exact realizations, we will pinpoint where canonical-correlations ideas can be exploited to extend our preliminary development here, for addressing the reduced-order modeling problem.

3.8.1 Overall strategy and design of root node

As in our development in Sections 3.3 through 3.5, our modeling strategy here is focused on fulfilling the decorrelating role (2.3) of state information. However, we will no longer myopically seek the fulfillment of this condition; now the information content of, say, node s will be more closely tied to the information content at the children nodes $s\alpha_1$ and $s\alpha_2$.

In keeping with our basic strategy, the root-node state $x(0)$ should represent just enough information about the finest-scale process ξ_0 to ensure that conditioned on this information, the left half and the right half of the finest-scale process (i.e., $\xi_{0\alpha_1}$ and $\xi_{0\alpha_2}$) are uncorrelated. At this point, we recall the canonical correlation decomposition that we described in Corollary 1. We recall that Corollary 1 guarantees that we can decompose $\xi_{0\alpha_1}$ and $\xi_{0\alpha_2}$ as

$$\begin{pmatrix} \xi_{0\alpha_1} \\ \xi_{0\alpha_2} \end{pmatrix} = \begin{pmatrix} H_{0\alpha_1} \\ H_{0\alpha_2} \end{pmatrix} n_0 + \begin{pmatrix} \nu_{0\alpha_1} \\ \nu_{0\alpha_2} \end{pmatrix},$$

where the random vectors n_0, $\nu_{0\alpha_1}$ and $\nu_{0\alpha_2}$ are uncorrelated. The important point is that if we let $x(0) = n_0$, then $\xi_{0\alpha_1}$ and $\xi_{0\alpha_2}$ will be uncorrelated, conditioned on $x(0)$.

There is a nice, additional benefit of the choice $x(0) = n_0$. In particular, Corollary 1 guarantees that in the sense of Proposition 1, this choice yields the lowest possible state

dimension for the root node. Hence, we employ n_0 as the value for $x(0)$,

$$x(0) = n_0, \quad \text{and} \quad P(0) = E(n_0 n_0^T),$$

thereby maintaining consistency with our decorrelating strategy, with the most compact possible representation for the requisite information (in an exact realization).

For the purposes of the rest of this development, we note that the residuals $\tilde{\xi}_{0\alpha_1|0}$ and $\tilde{\xi}_{0\alpha_2|0}$ are given by

$$\tilde{\xi}_{0\alpha_i|0} = \nu_{0\alpha_i} \qquad (i = 1, 2),$$

and the least-squares prediction matrices $H_{0\alpha_1|0}$ and $H_{0\alpha_2|0}$ are given by

$$H_{0\alpha_i|0} = H_{0\alpha_i} \qquad (i = 1, 2).$$

3.8.2 Design of Intermediate-level Nodes

Now, let us consider all of the nodes that lie between the root node and the set of finest-scale nodes. We use an inductive-style argument to design the structure and information content of these nodes. There are two components of our inductive hypothesis. First, we assume that at all the nodes at levels $0, 1, ..., k$ (for some integer k, $1 \leq k < M$) the corresponding state vectors fulfill the decorrelating role (2.3) of state information. Second, we assume that we know the covariance of the zero-mean residual variable $\tilde{\xi}_{s|s\bar{\gamma}}$.

Let s denote any arbitrary node at level $k+1$. Just as with all of its ancestor nodes, $x(s)$ should represent just enough information to ensure that (2.3) is fulfilled. We conveniently decompose into three steps the determination of $x(s)$. In a rough sense, these three steps can be described as *maintenance* of previously generated information, *generation* of new information, and *consolidation* of these two types of information into a more compact form.

Maintenance of Ancestor Information

In the first step, we carry down all the information from the parent node $s\bar{\gamma}$. This information (or at least a portion of it) will be necessary to maintain the conditional uncorrelatedness that we inductively assume was established at the parent node; in particular, we must maintain the property that ξ_s and ξ_{s^c} be uncorrelated, conditioned on $x(s)$. Thus, we tentatively let

$$x'(s) = x(s\bar{\gamma}),$$

where we have appended a prime to $x(s)$, as a reminder that this choice is only tentative.

Generation of New Information

In the second step, we augment $x'(s)$ with additional information, chosen to be sufficient to guarantee fulfillment of (2.3) at the given node. The specific nature of this additional information is again determined by using a canonical correlation decomposition. Appealing to Corollary 1, we decompose the two residual variables $\tilde{\xi}_{s\alpha_1|s\bar{\gamma}}$ and $\tilde{\xi}_{s\alpha_2|s\bar{\gamma}}$ as

$$\begin{pmatrix} \tilde{\xi}_{s\alpha_1|s\bar{\gamma}} \\ \tilde{\xi}_{s\alpha_2|s\bar{\gamma}} \end{pmatrix} = \begin{pmatrix} H_{s\alpha_1} \\ H_{s\alpha_2} \end{pmatrix} n_s + \begin{pmatrix} \nu_{s\alpha_1} \\ \nu_{s\alpha_2} \end{pmatrix},$$

where n_s, $\nu_{s\alpha_1}$ and $\nu_{s\alpha_2}$ are uncorrelated. As we presently verify, n_s is the augmenting information that we seek, and hence, we modify $x'(s)$ to become

$$x'(s) = \begin{pmatrix} x(s\bar{\gamma}) \\ n_s \end{pmatrix}. \tag{3.51}$$

We retain the prime on $x(s)$ as a reminder that this choice is still tentative.

To show that our modified $x'(s)$ does indeed fulfill its decorrelating role (2.3), we first note that by induction, ξ_s and ξ_{s^c} are uncorrelated, conditioned on $x(s\bar{\gamma})$. Furthermore,

$$\begin{aligned} E\left[\xi_{s\alpha_1} \mid \xi_{s\alpha_2}, x'(s)\right] &= E\left\{E\left[\xi_{s\alpha_1} \mid x(s\bar{\gamma})\right] + \tilde{\xi}_{s\alpha_1|s\bar{\gamma}} \mid \tilde{\xi}_{s\alpha_2|s\bar{\gamma}}, x(s\bar{\gamma}), n_s\right\} \\ &= E\left[\xi_{s\alpha_1} \mid x(s\bar{\gamma})\right] + E\left(\tilde{\xi}_{s\alpha_1|s\bar{\gamma}} \mid n_s\right), \end{aligned}$$

and hence, $\xi_{s\alpha_1}$ and $\xi_{s\alpha_2}$ are uncorrelated, conditioned on $x'(s)$. Combining this result with the conditional uncorrelatedness of ξ_s and ξ_{s^c}, we find that (2.3) is fulfilled.

Consolidation of Information

In the third step, we consolidate the information content of $x'(s)$ into a more compact form. Although, in its current form, the vector $x'(s)$ is consistent with our basic strategy, it may contain redundant and/or superfluous information that unnecessarily burdens its dimension. To see that this possibility is genuine, let us recall the Brownian motion construction in Section 2.3. At the left node of the second level in this construction, we implicitly discard the root node information about the value of the right end-value of the Brownian motion process; in this way, we have obtain a more compact representation of the needed information, without any effect on the finest-scale statistical behavior.

For Markov and reciprocal processes, there is no ambiguity about what information we can discard; on the other hand, in general, we require systematic tools for carrying out this task. We now describe an effective technique for reducing the information in $x'(s)$ to its essential component. This technique is based on the following lemma.

Lemma 1 *Let $x'(s)$ be a state vector that satisfies (2.3). Let $x(s)$ be a state vector that is related to $x'(s)$ in the following sense:*

$$E\left(\xi_s \mid x(s)\right) = E\left(\xi_s \mid x'(s)\right).$$

This condition is sufficient to guarantee that $x(s)$ also satisfies (2.3).

Proof: We prove the lemma in two stages. First, we show that ξ_s and ξ_{s^c} are uncorrelated, conditioned on $x(s)$:

$$\begin{aligned} E\left(\xi_s \mid x(s), \xi_{s^c}\right) &= E\left[E\left(\xi_s \mid x(s)\right) + \tilde{\xi}_{s|s} \mid x(s), \xi_{s^c}\right] \\ &= E\left[\xi_s \mid x(s)\right] + E\left(\tilde{\xi}_{s|s} \mid \xi_{s^c}\right) \\ &= E\left(\xi_s \mid x(s)\right). \end{aligned}$$

Second, we show that $\xi_{s\alpha_1}$ and $\xi_{s\alpha_2}$ are uncorrelated, conditioned on $x(s)$:

$$E\left(\xi_{s\alpha_1} \mid x(s), \xi_{s\alpha_2}\right) = E\left[E\left(\xi_{s\alpha_1} \mid x(s)\right) + \tilde{\xi}_{s\alpha_1|s} \mid \xi_{s\alpha_2}, x(s)\right]$$

$$\begin{aligned} &= E\left[\xi_{s\alpha_1} \mid x(s)\right] + E\left(\tilde{\xi}_{s\alpha_1|s} \mid \xi_{s\alpha_1}\right) \\ &= E\left(\xi_{s\alpha_1} \mid x(s)\right), \end{aligned}$$

thereby establishing the lemma. **QED**.

In order to exploit the lemma to compress $x'(s)$, we begin by expressing $E[\xi_s \mid x'(s)]$ as a function of $x(s\bar{\gamma})$ and n_s, thereby highlighting the roles of these variables. We have

$$\begin{aligned} E\left[\xi_s \mid x'(s)\right] &= E\left[E\left(\xi_s \mid x(s\bar{\gamma})\right) + \tilde{\xi}_{s|s\bar{\gamma}} \mid x(s\bar{\gamma}), n_s\right] \\ &= H_{s|s\bar{\gamma}}\, x(s\bar{\gamma}) + E\left(\tilde{\xi}_{s|s\bar{\gamma}} \mid n_s\right) \\ &= H_{s|s\bar{\gamma}}\, x(s\bar{\gamma}) + \begin{pmatrix} H_{s\alpha_1} \\ H_{s\alpha_2} \end{pmatrix} n_s \\ &= \left[\, H_{s|s\bar{\gamma}} \quad \begin{pmatrix} H_{s\alpha_1} \\ H_{s\alpha_2} \end{pmatrix} \right] \begin{pmatrix} x(s\bar{\gamma}) \\ n_s \end{pmatrix}. \end{aligned} \tag{3.52}$$

Now comes an essential step. By QR factorization,

$$\begin{aligned} \left[\, H_{s|s\bar{\gamma}} \quad \begin{pmatrix} H_{s\alpha_1} \\ H_{s\alpha_2} \end{pmatrix} \right] &= \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix} \\ &= Q_1 R, \end{aligned} \tag{3.53}$$

where $(Q_1\ Q_2)$ is an orthonormal matrix, the columns of Q_1 span the column space of the left-hand side of (3.53) and and R is an upper-triangular matrix. In terms of this factorization, we can define a matrix $x(s)$ that satisfies the condition of Lemma 1 and has a dimension no larger than the dimension of $x'(s)$. Specifically, we let

$$\begin{aligned} x(s) &= R\, x'(s) \\ &= R \begin{pmatrix} x(s\bar{\gamma}) \\ n_s \end{pmatrix}. \end{aligned} \tag{3.54}$$

By combining this with (3.52) and (3.53), we can verify that $x(s)$ satisfies Lemma 1:

$$\begin{aligned} E\left[\xi_s \mid x'(s)\right] &= Q_1\, R\, x'(s) \\ &= Q_1\, x(s) \\ &= E\left[\xi_s \mid x(s)\right]. \end{aligned}$$

Thus, we employ (3.54) as our definition of $x(s)$.

There are several consequences of this choice for $x(s)$. First, this choice implicitly provides us with appropriate values for $A(s)$ and $B(s)$. In particular, to be consistent with (3.54), we must let

$$A(s) = R_1, \qquad B(s) = \left[R_2\, E(n_s n_s^T)\, R_2^T\right]^{1/2}. \tag{3.55}$$

In this expression, the matrices R_1 and R_2 constitute a block partition of R,

$$R = \begin{pmatrix} R_1 & R_2 \end{pmatrix},$$

in which the number of columns in R_1 is equal to the dimension of $x(s\bar{\gamma})$. Our choice for $x(s)$ also implies that the residuals $\tilde{\xi}_{s\alpha_1|s}$ and $\tilde{\xi}_{s\alpha_2|s}$ are given by

$$\tilde{\xi}_{s\alpha_i|s} = \nu_{s\alpha_i} \qquad (i = 1, 2).$$

Finally, the least-squares prediction matrix $H_{s|s}$ is given by

$$H_{s|s} = Q_1.$$

We have now specified everything needed for $x(s)$, $A(s)$, and $B(s)$, and we have maintained consistency with the inductive hypotheses. Thus, the induction can continue.

3.8.3 Design of Finest-Scale Nodes

At the finest scale, our approach changes slightly. We are no longer interested in fulfilling (2.3), because that condition no longer makes sense. Instead, we want to simply let

$$x(s) = H_{s|s\bar{\gamma}}\, x(s\bar{\gamma}) + \tilde{\xi}_{s|s\bar{\gamma}}.$$

Consequently, we let

$$\begin{aligned} A(s) &= H_{s|s\bar{\gamma}}, \\ B(s) &= \left[E\left(\tilde{\xi}_{s|s\bar{\gamma}}\tilde{\xi}^T_{s|s\bar{\gamma}}\right)\right]^{1/2}. \end{aligned}$$

3.8.4 Final Comments

We have now completely described our realization procedure. In doing so, we have established both that a solution exists and that one can, in principle, be found. One of the principal difficulties with actually implementing this procedure is that we must keep track of the covariance matrices of the random vectors $\tilde{\xi}_{s|s}$; for problems of practical size, both the calculation and storage of these matrices is prohibitive. There may be merit in seeking ways to combat this computational burden, especially in modeling situations in which it is imperative to propagation coarse-scale information in a consistent way.

3.9 Conclusion

We have developed elements of a theory for multiscale stochastic realization. We have focused in particular on the problem of building multiscale models to realize, either exactly or approximately, prespecified finest-scale statistics. In this context, we have formalized the reduced-order modeling problem, we have developed model-building algorithms for addressing this problem, and we have demonstrated the practicality of our approach in an extensive set of numerical experiments. Finally, we have noted some non-trivial differences between time-series stochastic processes and multiscale stochastic processes.

Chapter 4

A Multiresolution Approach to Discriminating Targets from Clutter in SAR Imagery

In most detection and estimation problems, there is no ready availability of a complete statistical description of the quantities relevant to the problem, and thus, in these cases, the model-building techniques of the previous chapter are not directly applicable. Instead, we must build an appropriate multiscale model from the observed data directly. In this chapter, we consider an important problem in automatic target recognition (ATR), for which we must apply so-called techniques of *multiscale model identification.*

4.1 Introduction

The fundamental ATR problem is to detect and recognize objects of interest (i.e., *targets*) in a noisy environment (i.e., *clutter*) that has been imaged by an imperfect sensor. The heart of an ATR system is an integrated collection of algorithms designed to process sensor measurements so that the targets can be efficiently detected and identified. These algorithms are applied on a computer and ideally, they are organized so that human intervention is not required. In a military context, the hope is that if computers can be made to detect and recognize targets automatically, then the workload of a pilot can be reduced and the accuracy and efficiency of the pilot's weapons can be improved. We emphasize though that the applicability of ATR technology is not limited to the purview of military surveillance. For example, the technology can provide insight into the problem of recognizing landmarks sensed by a visual navigation system or by a robotic system. We direct the reader to [21] and the references therein for a thorough overview of the ATR problem.

A critical challenge within ATR is to determine automatically the locations and deployments of various kinds of military apparatus. Furthermore, in addressing this challenge, a desirable attribute of any ATR system is the ability to reject regions of exclusively natural clutter in a computationally fast and simple way. Resources can then be focused on the classification of a relatively small number of regions of interest containing man-made objects that are potentially targets.

In this chapter, we consider ATR for the case of a system whose inputs are synthetic-

aperture radar (SAR) images.[1] Within this problem domain, we both develop and extensively test a new algorithm for discriminating man-made objects from natural clutter. The novel feature of our approach is its exploitation of the characteristically distinct variations in speckle pattern, for imagery of natural clutter and of man-made objects, as image resolution is varied from coarse to fine. The fact that speckle has multiresolution characteristics is also noted and exploited in [61]. However, in contrast to that work, where the different characteristics of natural clutter and man-made objects are used to analyze individual image pixels, in this paper we use our multiscale framework to model and exploit these characteristics over entire blocks of imagery.

To understand the nature of the multiresolution characteristic of SAR imagery, we begin by recalling that SAR is a *coherent* sensing device. This coherence implies that the complex-valued radar reflectivity measured in any given resolution cell of a 2-D SAR image is equal to the coherent summation of all the returns from the scatterers residing in that resolution cell. Furthermore, as resolution is changed, the value of this coherent summation changes, due to migration of scatterers either into or out of the resolution cell. These elementary observations lead us to the most important point: the variation in speckle pattern, as a function of resolution, is typically different for natural clutter and for man-made objects. In simplified terms, the reason is that for natural clutter, there is typically a large number of equivalued scatterers in a resolution cell, while for man-made objects, there is typically only a small number of prominent scatterers [61]. This difference leads to very different statistics for the variation in speckle pattern as resolution changes.

This description of the SAR scattering mechanism suggests that there is considerable information in the *phase* of the complex-valued reflectivity measurements. Specifically, we can use the phase information to create coherently a sequence of images, each image having successively coarser resolution; in turn, this sequence of images can be analyzed collectively to discriminate between natural clutter and man-made objects. This coherent approach stands in contrast to typical SAR ATR algorithms, where phase information is discarded; in these cases, the complex-valued SAR imagery is converted to magnitude (or log-magnitude) form before ATR processing is applied. Our approach is more in line with the increasing realization in the SAR community that there may be merit in exploiting the phase information in complex data. Several potential uses of this information have been identified: (i) to capture the multiresolution characteristics of speckle, as considered here and in [61], (ii) to capture aspect-varying characteristics of radar cross-section, as considered in [10] and [4], and (iii) to manage the phase distortion induced by target motion. While the focus of this paper is on the first of these, we believe that the statistical foundations established here may very well be useful in these other contexts as well.

Although previous multiscale modeling work provide motivation for considering the possibility of multiscale representations of SAR imagery, we emphasize that none of that previous work has direct applicability to our problem. We identify two such reasons, in order to both clarify the role of previous work and highlight outstanding obstacles. First, the focus in [43] and in the previous chapter was on designing multiscale dynamics (i.e., principally, choosing values for $A(s)$ and $B(s)$) so that a desired, prespecified statistical structure emerged at the *finest* scale, with no detailed regard for the structure of the coarser scales. Consequently, no mechanism was provided for explicitly embedding non-local process information as coarse-scale states. Second, both [43] and the previous chapter were exclusively

[1] A comprehensive reference on synthetic-aperture radar is provided by [16]. A nice overview, tailored to the typical background of the signal processing community, is provided by [51].

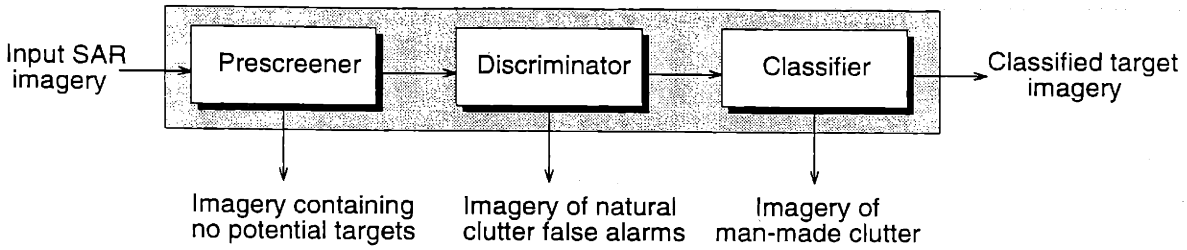


Figure 4-1: Illustration of the input-output operation of the SAR ATR system described in [52]. The input consists of SAR imagery representing many square kilometers of terrain and potentially containing several targets of interest; the output consists of locations and classification labels for these targets. This article describes a novel, multiresolution-based approach to the discrimination done in the second stage.

concerned with building models from a complete statistical description, rather than from data alone. Both of these issues will be addressed in this chapter.

We build a pair of multiscale stochastic models for SAR imagery: one model capturing the statistical characteristics in the scale-to-scale variations in SAR imagery of natural clutter and a corresponding one for imagery of man-made objects. Interestingly, the method we use for constructing these models from actual data is the direct scale-recursive extension of a widely used method of autoregressive modeling for time series. We validate our models on actual SAR data and then use them to define a multiresolution discriminant. This discriminant is the likelihood ratio for distinguishing between natural clutter and man-made objects, given a multiresolution sequence of SAR images. Thanks to the structure of our models, the calculation of these likelihoods is a computationally simple task [44].

We incorporate our multiscale stochastic models and the resulting multiresolution likelihood discriminant into an existing SAR ATR system developed at Lincoln Laboratory [40, 52]. This system has been designed to operate in an off-line, experimental setting; it has been rigorously tested over the past several years, and is one of the first systems of its kind to process large quantities of actual SAR data. The system is conveniently decomposed into a sequence of three processors: a prescreener, a discriminator and a classifier (see Figure 4-1). The prescreener searches through imagery representing many square kilometers of terrain, and outputs a collection of so-called regions of interest[2] (ROIs) centered at possible target locations. The discriminator applies further processing to distinguish between two kinds of ROIs: those containing man-made objects and those containing natural clutter. All ROIs that appear to contain natural clutter are discarded. Finally, the classifier assigns each remaining ROI to a predefined target category, or to a *none-of-the-above* category if the ROI appears to contain man-made clutter.

Our multiscale discriminant fits naturally into the second or discrimination stage of the Lincoln Laboratory ATR system. In an idealized setting, this discriminant would be a sufficient statistic for making the hypothesis testing decision [63]. However, a more practical, realistic view is that our stochastic models do not capture all of the characteristics that distinguish man-made objects from natural clutter. We take this latter view, and in conjunction, we take advantage of the years of development that have gone into the Lincoln Laboratory ATR system, which have led to the identification of a number of useful characteristics for carrying out discrimination [40]. Guided by this previous work, we first develop an optimized version of Lincoln's discriminator, in which we measure and exploit a small

[2]Each region of interest (ROI) is a subimage extracted from the original SAR data set; collectively, all ROIs represent only a small fraction of this original data set.

number of size and brightness characteristics of the ROI. Then, we combine the resulting measured values with the value of the multiresolution likelihood ratio into a single scalar-valued measure of the so-called distance of the ROI from the class of targets of interest. In effect, then, we treat the likelihood ratio as a multiresolution-based textural feature. All ROIs having a large distance are labeled *non-target* and the remaining ROIs are labeled *target*.

We have applied our new discrimination algorithm to an extensive data set of 0.3-meter resolution, HH polarization[3] imagery gathered with the Lincoln Laboratory millimeter-wave SAR [31]. As we show in Section 4.4, the detection results are impressive. In particular, we demonstrate a substantial and statistically significant improvement in the receiver operating characteristics when we augment our optimized version of Lincoln's standard discriminator with our new multiresolution discriminant. This result is surprising good, in light of the number of years over which the standard discriminator has been developed and refined and the relatively simple multiresolution algorithm we have used here. The result conclusively demonstrates that multiresolution methods have an effective and important role to play in SAR ATR algorithms.

This chapter is divided into four major sections. In the next we describe our procedure for identifying multiscale models for SAR imagery. We then develop our multiresolution discriminant, and describe both the standard discriminator and our refinement of it. Next, we show the results of our extensive testing of these discriminators. Finally, we summarize the main points of the paper and suggest directions for future work in SAR applications of multiresolution-based techniques.

4.2 Identification of Multiscale Models for SAR Imagery

In this section, we develop our multiscale stochastic models for SAR imagery. These models are particular representatives of the multiscale model class introduced in Chapter 2, suitably specialized to characterize the statistical distribution of speckle pattern variation in a multiresolution sequence of SAR images.

In light of the vastness of the two classes *natural clutter* and *man-made objects*, there is certainly an issue regarding the number of multiscale models we should build. One could imagine developing a number of models for our designated discrimination application. We could develop a whole suite of models for natural clutter, including one for grass, another for trees, and so forth, and a whole separate suite of models for man-made objects. However, for this initial development and demonstration, we choose to develop only two models, with a single model representing each class. Our model for natural clutter, hereinafter referred to as our *natural-clutter model*, is specifically designed to describe imagery of grass, while our model for man-made objects, hereinafter referred to as our *man-made model* is specifically designed to describe imagery of tactical targets. A natural question is whether the resulting models lead to a discriminant that is robust to variations within each of the two large classes. We will see in Section 4.4 that the answer is yes.

Our development proceeds as follows. We begin with a detailed description of the objects we wish to model and classify; in particular, we describe our procedure for generating multiresolution sequences of images. Next we describe our approach to multiresolution

[3]The nomenclature "HH polarization" means that the SAR sensor both transmits and receives electromagnetic radiation in which the electric field has a horizontal orientation with respect to the ground plane.

model identification. This approach has three principal steps for each of the models we identify. First, we restrict the search to a simply parameterized subclass of multiscale stochastic models, namely linear autoregressions in scale. Second, we choose appropriate regression coefficients, based on a simple optimization criterion. Finally, we characterize the statistical distributions of the model driving noise w_s.

4.2.1 Generation of Multiscale Image Sequences

Our procedure for creating a multiresolution sequence of images begins with complex-valued SAR imagery, formed to the highest resolution available. Each pixel value in this imagery represents a measurement of both the amplitude and phase of the radar reflectivity of the scatterers within a resolution cell. In all of our work, we use HH polarization imagery gathered with the Lincoln Laboratory millimeter-wave SAR [31]. Although this choice will affect the specifics of the models we build, our general procedures should be more broadly applicable.

From this full resolution imagery, we assume that ROIs have been extracted. For each ROI $\mathcal{I}$, we create a multiresolution sequence of images, $I_0, I_1, \ldots, I_L$. This sequence is created directly from $\mathcal{I}$, with no dependence on the rest of the SAR data set from which $\mathcal{I}$ was extracted.

To prepare for our detailed description of the processing used to go from $\mathcal{I}$ to $I_0, I_1, \ldots, I_L$, we introduce some useful notation and conventions. We assume for simplicity that the image $\mathcal{I}$ has the same resolution in both range and cross-range, and we denote this resolution by δ (for the Lincoln Laboratory millimeter-wave SAR, $\delta = 0.3$ meters). We denote by $\mathcal{I}(k,l)$ the measured reflectivity at range/cross-range position (k,l). Finally, for convenience only, we assume that this image array is square, consisting of $N \times N$ pixels, where $N = 2^M$ for some integer M.

Finest-scale image

The finest-scale image is created by applying to $\mathcal{I}$ log-detection (defined in (4.1)), followed by normalization. The resulting image is denoted by I_0; it has resolution $\delta \times \delta$ and consists of $N \times N$ pixels.

The intermediate, log-detected image I_0' is defined by

$$I_0'(k,l) = 20 \log_{10} |\mathcal{I}(k,l)|. \tag{4.1}$$

Our use of log-detection is motivated by standard practice in the SAR community, where the dB format has an established history. The logarithm effectively compresses the radar-reflectivity variation, which typically spans several orders of magnitude, so that it is easier to interpret visually. Furthermore, many established SAR ATR algorithms call for dB formatting (see, for example, [28]); one reason is that the logarithm operation converts the multiplicative effect of speckle noise to an additive effect, which is sometimes considered easier to analyze.

For carrying out hypothesis test decisionmaking, however, there is a difficulty with directly using the log-detected image I_0'. The problem is that its pixel values are dependent on the radar sensor's absolute calibration, which is susceptible to spurious fluctuations. To eliminate this dependence, we apply a simple normalization to I_0', thereby yielding I_0.

Specifically, I_0' and I_0 are related by

$$I_0(k,l) = I_0'(k,l) - C_0,$$

where C_0 is equal to the sample mean of I_0',

$$C_0 = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} I_0'(k,l). \tag{4.2}$$

Because I_0 has no dependence on absolute calibration, our decisionmaking exploits only the *relative* variation of image intensity, with respect to a mean level of zero.

Coarse-scale images

We denote the coarser-scale images by $I_1, I_2, \ldots, I_L$, respectively. We form these images from the original complex, fine-scale data $\mathcal{I}$ by sequentially applying three processing steps: (i) lowpass filtering, (ii) decimation, and (iii) log-detection with normalization. For image I_m, this processing ultimately yields a $2^{-m}N \times 2^{-m}N$ square array image having resolution $2^m\delta \times 2^m\delta$.

The first processing step effectively reduces both the bandwidth of each SAR pulse (thus coarsening range resolution) and the width of the SAR aperture (thus coarsening cross-range resolution). To describe this operation, we denote the inverse discrete Fourier transform of the 2-D complex data $\mathcal{I}$ by $\tilde{\mathcal{I}}$, where

$$\tilde{\mathcal{I}}(p,q) = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} \mathcal{I}(k,l)\exp\left(j\frac{2\pi k}{N}p\right)\exp\left(j\frac{2\pi l}{N}q\right).$$

In terms of $\tilde{\mathcal{I}}$, we define I_m' via

$$I_m'(k,l) = \sum_{p=0}^{N-1}\sum_{q=0}^{N-1} \tilde{\mathcal{I}}(p,q)H_m(p)H_m(q)\exp\left(-j\frac{2\pi k}{N}p\right)\exp\left(-j\frac{2\pi l}{N}q\right),$$

where $H_m(p)H_m(q)$ represents a separable 2-D Hamming window with $H_m(p)$ defined to be

$$H_m(p) = \begin{cases} 0.54 + 0.46\cos\frac{2\pi p}{N} & 0 \le p < 2^{m-1} \;\&\; N - 2^{m-1} \le p < N \\ 0 & 2^{m-1} \le p < N - 2^{m-1} \end{cases}.$$

In the second processing step, we eliminate the oversampling in I_m' by decimating by a factor of 2^m in both range and cross-range. The result is denoted by I_m'' and is related to I_m' by

$$I_m''(k,l) = I_m'\left(2^m k, 2^m l\right), \quad 0 \le k,l < 2^{-m}N.$$

Finally, in the third step, we apply log-detection and normalization. These operations have exactly the same form and rationale as the detection and normalization operations we

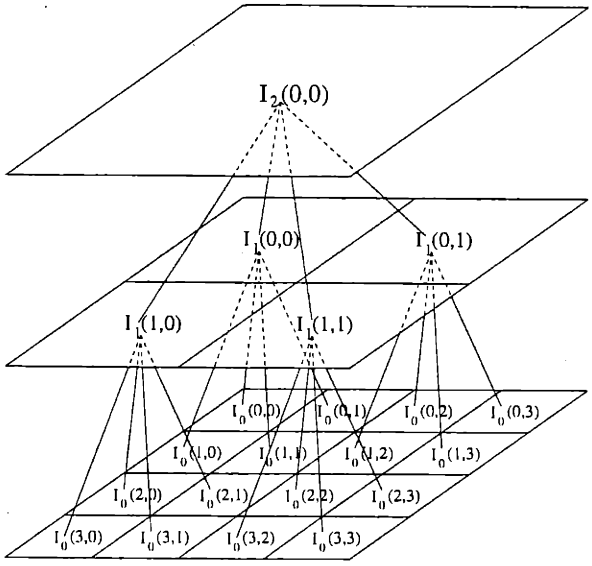


Figure 4-2: Illustration of a multiresolution sequence of three SAR images, together with the quadtree onto which we map pixel values. In this example, the pixel value at scale m and position (k,l) is denoted by $I_{2-m}(k,l)$.

applied to $\mathcal{I}$ to yield I_0. The image $I_m(k,l)$ is thus related to I''_m via

$$I_m(k,l) = 20\log_{10}|I''_m(k,l)| - \frac{1}{(2^{-m}N)^2}\sum_{p=0}^{(2^{-m}N-1)}\sum_{q=0}^{(2^{-m}N-1)} 20\log_{10}|I''_m(p,q)|.$$

Mapping the multiscale SAR image sequence onto a quadtree

The multiresolution image sequence $I_0, I_1, \ldots, I_L$ is matched quite naturally to the structure of a quadtree, and we consequently use the quadtree for all our SAR image modeling. In Figure 4-2, we illustrate our convention for the correspondence between pixel values and tree nodes. To formalize this convention, we associate each node s on the quadtree with a 3-tuple (m,k,l), where m denotes scale and (k,l) denotes 2-D location; correspondingly, we denote by $I(s)$ the image pixel residing at node s, namely $I_m(k,l)$. For example, in the context of Figure 4-2, $I(0)$ corresponds to $I_2(0,0)$, and $I(0\alpha_i)$ $(i=1,2,3,4)$ corresponds to $I_1(0,0), I_1(0,1), I_1(1,1)$ and $I_1(1,0)$, respectively. In a manner that we will make precise in the next section, we treat these pixel values $I(s)$ as our multiscale process observations $y(s)$.

Our example in Figure 4-2 illustrates a special case in which we have formed a complete sequence of images, down to a single-pixel image at the coarsest resolution possible (i.e., $L=M$). More generally, we allow the possibility of truncating the image sequence at some image having more than a single pixel (i.e., $L<M$). When this possibility occurs, the nodes at the coarser scales of the tree have no corresponding measurements. This additional flexibility is useful, because beyond a certain coarseness of resolution, SAR imagery conveys very little meaningful information.

4.2.2 Identifying the Multiscale Dynamics

We are now prepared to turn our attention from the details of *generation* of multiresolution image sequences to the details of *statistical characterization* of these sequences. Specifically, we wish to characterize the joint statistical distribution of pixel values in $I_0, I_1, \ldots, I_L$. Given our defined relationship between pixels in the SAR images and nodes on the quadtree, the heart of this characterization is to determine multiscale dynamics that yield process statistics consistent with our process observations. This characterization must ultimately yield, for each of the stochastic models we build, suitable values for several model parameters: (i) the matrices $A(s), B(s)$ and $C(s)$, (ii) the distribution of the driving noise $w(s)$ and (iii) the distribution of the initial condition $x(0)$.

We remark that although there is a considerable body of literature devoted to statistical modeling of SAR imagery, this work has typically focused on characterizing marginal, single-pixel statistics of imagery at a single resolution [36, 68]. In contrast, our interest lies in jointly characterizing the scale-to-scale statistical coupling of a sequence of SAR images $I_0, I_1, \ldots, I_L$ spanning multiple resolutions. In principle, this desired characterization could be devised by combining a first-principles model for $\mathcal{I}$ together with the effect of the chain of processing steps from $\mathcal{I}$ to $I_0, I_1, \ldots, I_L$. However, given the state of our understanding of the imaging physics, this first-principles approach is problematic. As an alternative, we pursue a more purely statistical approach: we begin by specifying a parametric subset of the multiscale model class, and then we use so-called training data to complete the model specification, using an approach that is the direct scale-recursive quadtree extension of a well-known modeling technique for time-series.

Restriction to linear autoregression model class

We focus on a specific class of multiresolution models of the form (2.1). This class is motivated by the idea that in going from coarser to finer scales, the SAR image value $I_m(k,l)$ can be partially predicted using its coarser-scale ancestors, but that the image value also has an unpredictable component, due to the changing effect of speckle as resolution is varied. This line of reasoning is also pursued in [61] for the analysis of the multiscale characteristics of a single pixel. In terms of the quadtree picture in Figure 4-2, this single-pixel analysis corresponds, roughly, to separate modeling for the sequence of pixel values starting from each individual finest-scale pixel and proceeding upwards through the pixels at successive ancestor nodes. In what we now describe, we use a single, overall self-consistent statistical model for the *entire* sequence of multiresolution images represented on the quadtree. In particular, the physical interpretation of the multiscale effect of speckle suggests a model in which the SAR pixel value residing at node s (i.e., $I(s)$) is related to its ancestors by a linear autoregression in scale:

$$I(s) = a_{1,m(s)} I(s\bar{\gamma}) \; + \; a_{2,m(s)} I(s\bar{\gamma}^2) \; + \; \cdots \; + a_{R,m(s)} I(s\bar{\gamma}^R) \; + w(s). \tag{4.3}$$

This model's analogue in time-series analysis is extremely popular. This popularity is in part due to the simplicity of the autoregression, but is also due to its effectiveness in modeling a wide variety of phenomena and its successful use many applications [42]. These facts, together with our physical interpretation of the scale-varying effect of speckle, provide the motivation for our use here of the scale-recursive counterpart. Of course, we must validate that our resulting models are consistent with actual data; we will see that the models not only pass this consistency check, but are also quite effective in their designated

discrimination application.

There are several additional comments to make about our autoregressive model. With regard to (4.3), R is the order of the regression and $a_{1,m(s)}, a_{2,m(s)}, \ldots, a_{R,m(s)}$ are the scalar-valued regression coefficients. These regression coefficients are allowed to be scale-varying, but are restricted to be shift-invariant for any fixed scale. We will have occasion to refer collectively to the whole set of coefficients for a given scale, and for this purpose, we define the vector $\mathbf{a}_k$ as

$$\mathbf{a}_k \equiv \begin{pmatrix} a_{1,k} & a_{2,k} & \cdots & a_{R,k} \end{pmatrix}^T.$$

The term w_s represents the residual error in the prediction of $I(s)$. We assume that w_s and w_σ are statistically independent for $s \neq \sigma$. The probability distribution of w_s is allowed to be non-Gaussian, and furthermore the distribution is allowed to vary with scale. However, for any fixed scale, the distribution is assumed to be spatially invariant. We denote the standard deviation of the residuals at scale m by σ_m. As we will see, this independence assumption on w_s is what allows us to develop an extremely simple procedure for likelihood calculation for the entire piece of multiresolution imagery. Thus, the validation of this whiteness assumption is critical.

To accommodate the fact that we only have a finite number of SAR images, namely for the finest $L+1$ resolutions, we must statistically characterize the initial condition of the recursion in (4.3). This initial condition comprises the values of the pixels in the R coarsest-scale images $I_L, I_{L-1}, \ldots, I_{L-R+1}$. In devising a suitable characterization, we are guided by the following observation: we are more interested in the scale-to-scale variation of the speckle pattern than in the initial condition of this pattern. These relative interests lead us to impose no prior knowledge about the values of the R coarsest-scale images. We simply observe these images, and use the observed values as the initialization for the recursion in (4.3).

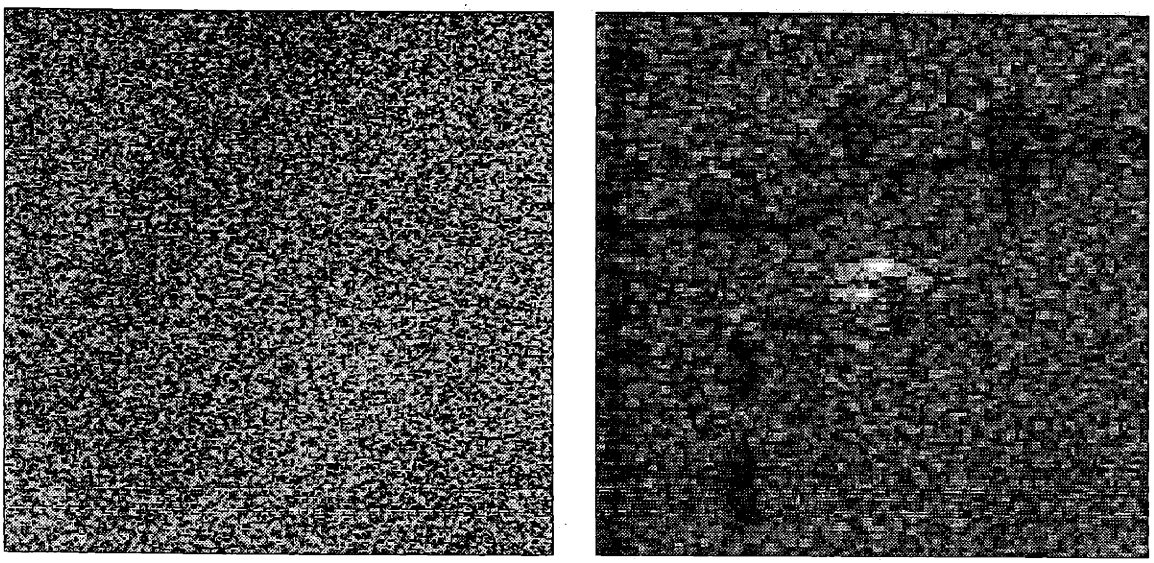
As a final remark, our linear autoregressive model can be expressed in a state-space form, exactly as in (2.1). We defer our development of this form until Section 4.2.2.

Identification of the regression coefficients

Given our specification of the class of linear autoregressions, our next task is to identify a regression order R and corresponding regression coefficients for each model we build. We identify these model parameters using complex-valued, training images, generated with the Lincoln Laboratory millimeter-wave SAR. For our natural-clutter model, we use a single image, which is displayed in Figure 4-3a. This image represents a homogeneous region of grass at $\delta \times \delta$ resolution[4], and consists of 256×256 pixels. Figure 4-3b depicts an example of a scene containing a man-made object. In contrast to the case of natural clutter (Figure 4-3a), the object in Figure 4-3b is spatially localized and nonstationary, and thus it makes no sense to build our model based on a large region of "homogeneous targets" analogous to Figure 4-3a. Thus, to build our target model, we use a training set of 64 SAR images (having 32×32 pixels each) of howitzers, each imaged at a different aspect angle.

We convert a given training image $\mathcal{I}$ into a multiresolution sequence of images $I_0, I_1, \ldots, I_L$, as described in detail in Section 4.2.1. We then systematically consider a sequence of possible regression orders, $R = 1, 2, 3$. For each proposed order, we apply the autoregression

[4] As defined in Section 4.2.1, $\delta = 0.3$ meters.



(a) (b)

Figure 4-3: Training SAR images used for model identification. These images were created using the Lincoln Laboratory millimeter-wave SAR. (a) Image is used to build our natural-clutter model; the image represents a homogeneous region of grass at $\delta \times \delta$ resolution. (b) Image of a target-like, man-made object at $\delta \times \delta$ resolution.

to the training data and solve for the regression coefficients that minimize the sum of the squares of the residuals. In particular, we implicitly define each regression vector $\mathbf{a}_m$ by the following relation:

$$\mathbf{a}_k = \arg \min_{\mathbf{a}_k \in \mathcal{R}^R} \left\{ \sum_{\{s;\, m(s)=k\}} \left[I(s) - a_{1,k} I(s\bar{\gamma}) - \ldots - a_{R,k} I(s\bar{\gamma}^R) \right]^2 \right\}. \qquad (4.4)$$

An explicit expression for $\mathbf{a}_k$ is straightforward to obtain from (4.4), as shown in Appendix C.1.

Our least-squares approach is the most widely used method for parameter estimation, and is eminently reasonable here, especially given our absence of prior knowledge about the statistics of the residuals $w(s)$ [42]. As we will see, the outcome of this procedure suggests both a natural model order and appropriate regression coefficients for each of the models we build.

Table 4.1 summarizes the outcome of applying our estimation procedure. The first column of the table lists the resolution of the image pixels to be predicted (i.e., the resolution of $I(s)$ in the autoregression in (4.3)). The second column lists the proposed order of the regression. Finally, the last two sets of columns list the regression coefficients $\mathbf{a}_m$ and the residual standard deviation σ_m for the natural-clutter and man-made models, respectively. For example, according to the table, the appropriate second-order regression for prediction of $2\delta \times 2\delta$ resolution pixels $I(s)$ in imagery of man-made objects is

$$\hat{I}(s) = 0.84 I(s\bar{\gamma}) - 0.16 I(s\bar{\gamma}^2),$$

with a residual standard deviation of $\sigma_m = 7.5$.

With regard to the natural-clutter model, the table suggests that there is no practical

Image resolution	*Model order*	*Natural-clutter model*				*Man-made model*			
		Regression coefficients			*Residual std. dev.*	*Regression coefficients*			*Residual std. dev.*
$\delta \times \delta$	1	0.28			5.4	0.70			7.2
	2	0.31	-0.011		5.4	0.67	0.10		7.0
	3	0.28	0.008	-0.01	5.4	0.69	0.12	0.008	7.0
$2\delta \times 2\delta$	1	0.30			5.3	0.87			7.6
	2	0.32	0.02		5.3	0.84	-0.16		7.5
	3	0.28	0.02	0.02	5.3	0.82	-0.11	0.009	7.5
$4\delta \times 4\delta$	1	0.25			5.5	0.58			8.5
	2	0.25	0.008		5.5	0.58	0.002		8.5
	3	0.25	-0.008	0.007	5.5	0.57	-0.009	0.01	8.5

Table 4.1: This table summarizes the outcome of our least-squares procedure for determining both model order R and regression coefficients $\mathbf{a}_m$. The first column lists the resolution of the image pixels to be predicted. The second column lists the proposed order of the regression. Finally, the last two sets of columns list the regression coefficients and the residual standard deviation for the natural-clutter and man-made models, respectively.

Image resolution	*Natural-clutter model*		*Man-made model*		
	Regression coefficient	*Residual std. dev.*	*Regression coefficients*		*Residual std. dev.*
$\delta \times \delta$	0.28	5.4	0.67	0.10	7.0
$2\delta \times 2\delta$	0.30	5.3	0.84	-0.16	7.5
$4\delta \times 4\delta$	0.25	5.5	0.58	0.002	8.5

Table 4.2: This table summarizes our final choices for both model order R and regression coefficient values $\mathbf{a}_m$ for each of our two models. The first column lists the resolution of the image pixels to be predicted. Then, the next two sets of columns list the regression coefficients and the residual standard deviation for the natural-clutter and man-made models, respectively.

benefit to using a model order greater than one; the higher-order regression coefficients have negligible magnitude and the standard deviation of the prediction error is not noticeably reduced by an increased model order. For these reasons, we use a first-order autoregression for the natural-clutter model. On the other hand, for the man-made model, a second-order regression appears to be preferable. In particular, the second-order regression coefficient is not negligible (at least for the $\delta \times \delta$ and $2\delta \times 2\delta$ images), and the standard deviation of the prediction error is reduced by increasing the model order from one to two. Because a third-order regression fails to continue this trend of increased benefit, we use a second-order autoregression for the man-made model. These model choices are summarized in Table 4.2, which represents a subset of Table 4.1.

As a final remark, we note that the leading regression coefficient in our man-made model is significantly larger than the sole regression coefficient in our natural-clutter model. In a loose sense, this observation implies that a multiresolution sequence of images of a man-made object is more tightly coupled than a corresponding sequence of images of natural clutter.

This interpretation is consistent with our description in the Introduction of the different SAR scattering mechanisms for natural clutter and man-made objects. In particular, we expect a multiresolution sequence of images of a man-made object to be tightly coupled, as the same few prominent scatterers dominate all the images. On the other hand, we expect a sequence of images of natural clutter to be only loosely coupled, as the large number of equivalued scatterers interfere with each other in a more unpredictable way as resolution is varied. In addition, the identification of a first-order autoregression for natural clutter and a second-order autoregression for targets is consistent with the individual-pixel models developed in [61] using simple theoretical models of the scene and the SAR imaging mechanism.

Validation of residual whiteness

As we have previously noted, a critical assumption in our modeling framework is that the residuals $w(s)$ in our models are statistically independent, both in space (for a fixed scale) and in scale. Whiteness in scale is, to a considerable degree, guaranteed by the nature of our fitting procedure in exactly the same way as for the case of time series analysis. In particular, the well-known principal of orthogonality provides theoretical assurance that when the expected error (4.4) is minimized, the resulting error in the scale-to-scale prediction (i.e., w_s in 4.3)) is uncorrelated with coarser scale features and thus with values $w(\sigma)$ at nodes σ that are ancestors of node s. This whiteness along paths of our quadtree in Figure 4-2 is, in fact, what is justified using a theoretical model, then exploited in [61]. However, in order to use our approach over an entire image, we want much more than this: we also want w_s to be white in *space* as well as in scale. To validate this spatial whiteness, we examine the sample correlation of the residuals that result when the appropriate autoregression is applied as a predictor to a multiresolution sequence of images.

An example of the data used in this validation for natural clutter is provided in Figure 4-4. The left column of this figure displays a multiresolution sequence of three images of the region of grass we used for training. Proceeding downward, the images have resolution $4\delta \times 4\delta$, $2\delta \times 2\delta$ and $\delta \times \delta$, respectively. The right column of the figure displays images of prediction residuals; in keeping with Table 4.2, the top image represents the residuals formed by the difference

$$I(s) - 0.3I(s\bar{\gamma}), \quad \text{with } I(s) \text{ in the image having } 2\delta \times 2\delta \text{ resolution,}$$

and the bottom image represents the residuals formed by the difference

$$I(s) - 0.28I(s\bar{\gamma}), \quad \text{with } I(s) \text{ in the image having } \delta \times \delta \text{ resolution.}$$

At least visually, Figure 4-4 suggests that the residuals are approximately uncorrelated. This is further confirmed by Figure 4-5, which displays the sample correlation function of the residual image from the lower-right corner of Figure 4-4. One can readily discern the impulse-like shape of this correlation function, which renders it in striking agreement with our model assumption. Although not shown, the same impulse-like shape is exhibited by the sample correlation functions of coarser-scale residuals. These numerical experiments are quite reassuring, and nicely demonstrate that our natural-clutter model can do a good job of decorrelating the prediction residuals across any fixed scale.

We now consider our man-made model. Proceeding in a manner parallel to our validation procedure for the natural-clutter model, we display in Figure 4-6 a collection of five images,

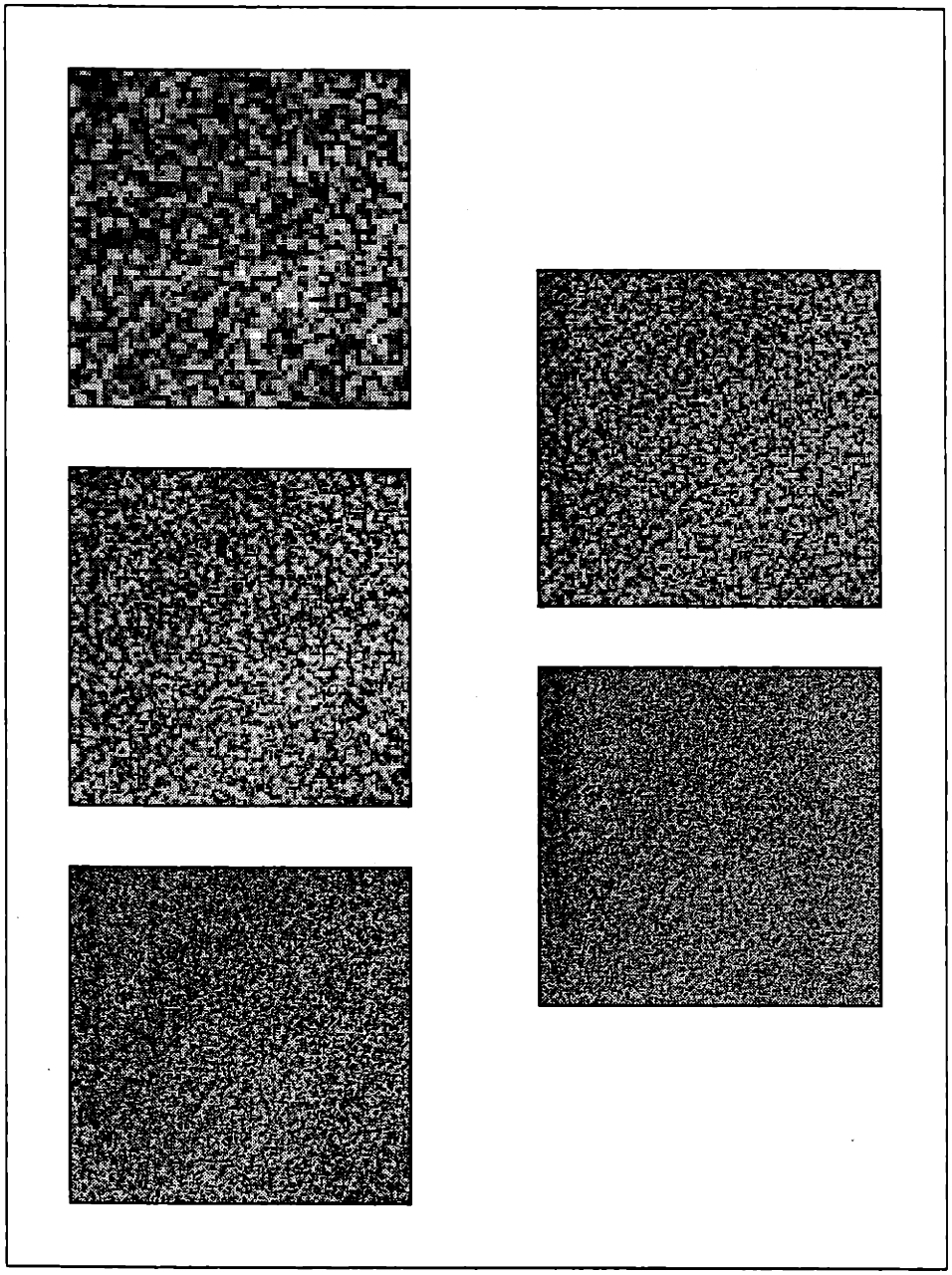


Figure 4-4: Images used to validate our natural-clutter model assumption that prediction residuals are white. The left column contains a multiresolution sequence of images of the region of grass we used for training; proceeding downward, the images have resolution $4\delta \times 4\delta$, $2\delta \times 2\delta$ and $\delta \times \delta$, respectively. The right column contains images of the prediction residuals; the top image represents $I(s) - 0.3I(s\bar{\gamma})$, with s in the image having resolution $2\delta \times 2\delta$ and the bottom image represents $I(s) - 0.28I(s\bar{\gamma})$, with s in the image having resolution $\delta \times \delta$. We note that the residuals appear approximately uncorrelated, in agreement with our model assumption.

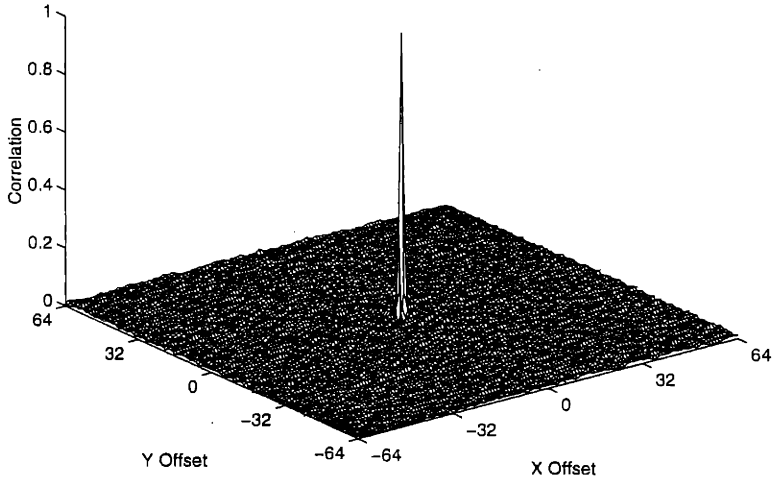


Figure 4-5: Sample correlation function for residuals in prediction of $\delta \times \delta$ resolution image of a region of grass, using only the $2\delta \times 2\delta$ resolution image. One can readily discern the impulse-like shape of this correlation function, which renders it in striking agreement with our model assumption.

three of which are SAR images and two of which are residual images. We immediately see that the residuals do not have the same completely uncorrelated appearance we observed in Figure 4-4. In this sense, the man-made model does not capture as completely the scale-to-scale statistical coupling of the multiresolution sequence of images. Nevertheless, as we will see in Section 4.4, if we ignore this remaining correlation and apply the resulting likelihood calculation methods based on the assumption of white residuals, we obtain excellent results. Of course, this also suggests that potentially even greater gains can be achieved if more sophisticated models are used, a point on which we comment further in Section 4.5.

Finally, for both of our models, we can measure the correlation of residuals across different scales. We have empirically found that the peak correlation between residuals at different scales is roughly 0.2, which is quite modest. In general, we conclude that the correlation of measured residuals behaves in manner impressively consistent with our model assumption of uncorrelatedness, particularly in the case of our natural-clutter model.

Identification of the residual distributions

Our final identification task is to characterize the probability distributions of the prediction residuals $w(s)$. For each model, we proceed by first calculating the sample cumulative distribution function (CDF) of the residuals associated with our training data (see Figures 4-4 and 4-6); then, we find a matching CDF that has a compact analytical form.

In Figures 4-7 and 4-8, we plot both empirical CDFs and our analytical fits to them; the first figure displays the entire CDFs, while the second focuses exclusively on the upper tails. We first consider the CDFs shown for the residuals associated with our man-made model. The sample CDF (i.e., the dashed line) summarizes the aggregate statistics of the residuals in the prediction of 136 finest-scale images of tactical targets. Each of these predictions is based on an autoregression using the coefficients listed in Table 4.2, applied to the coarser-scale images of the respective target. Our corresponding analytical fit (i.e., the dash-dot

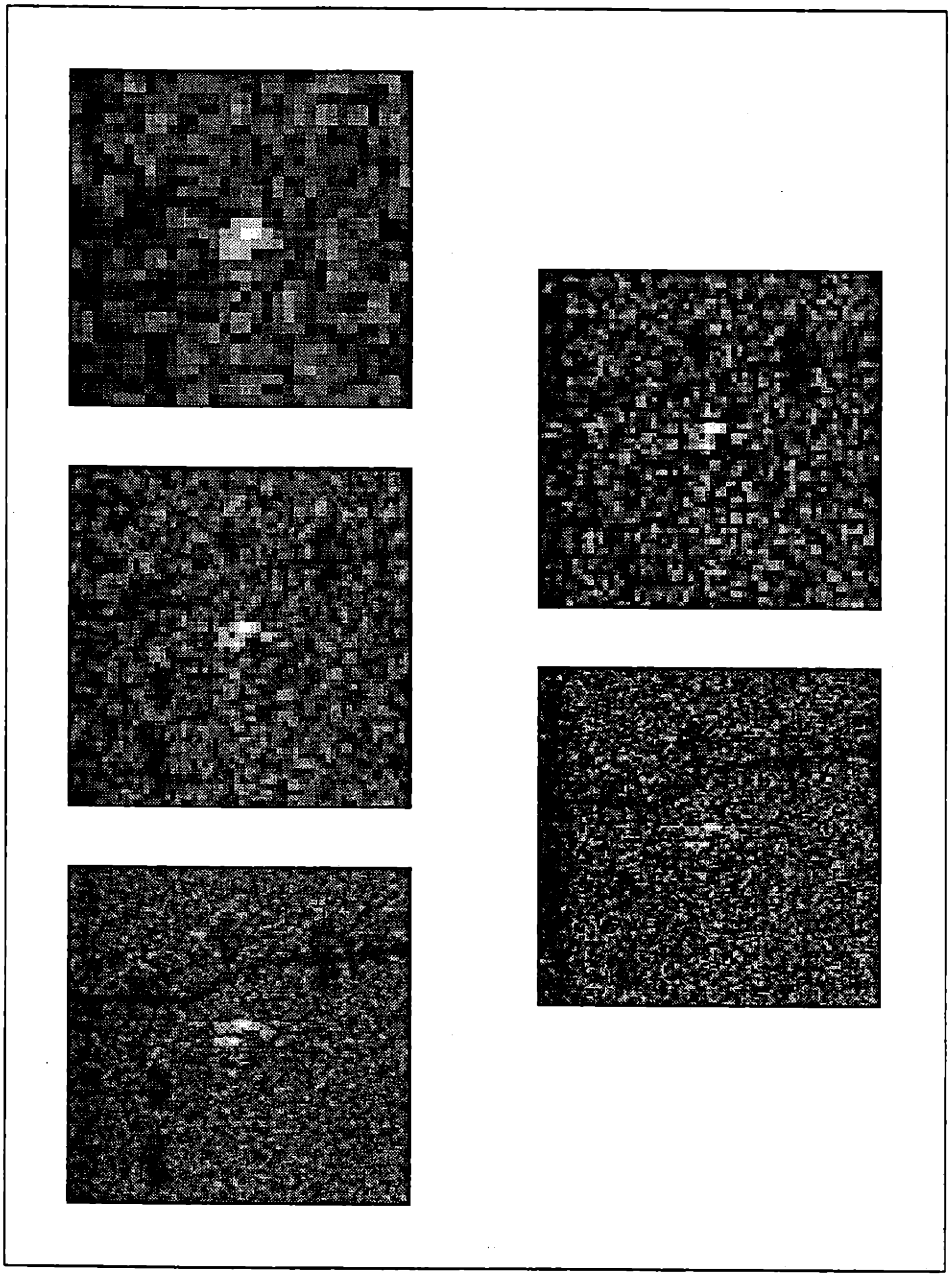


Figure 4-6: Images used to validate our man-made model assumption that prediction residuals are white. The left column contains a multiresolution sequence of images of a target-like object; proceeding downward, the images have resolution $4\delta \times 4\delta$, $2\delta \times 2\delta$ and $\delta \times \delta$, respectively. The right column contains images of the prediction residuals; the top image represents $I(s) - 0.84I(s\bar{\gamma}) + 0.16I(s\bar{\gamma}^2)$, with s in the image having resolution $2\delta \times 2\delta$ and the bottom image represents $I(s) - 0.67I(s\bar{\gamma}) - 0.1I(s\bar{\gamma}^2)$, with s in the image having resolution $\delta \times \delta$.

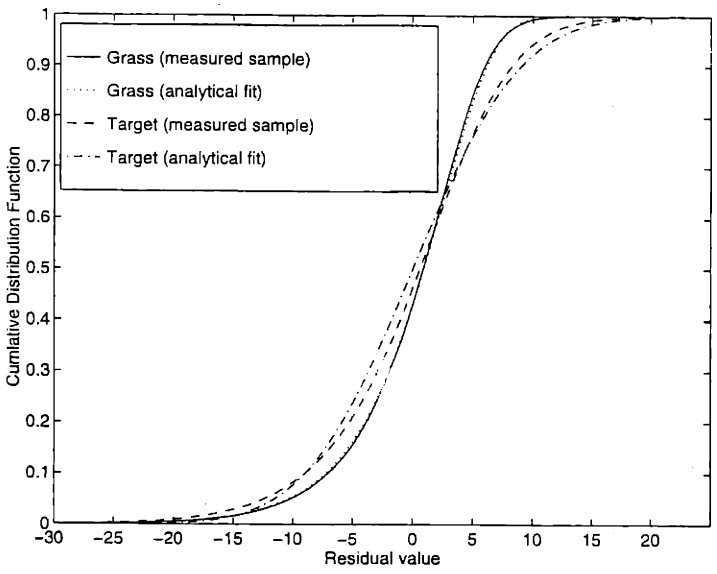


Figure 4-7: CDFs for prediction residuals associated with both our man-made model and our natural-clutter model. The sample CDF associated with grass summarizes the distribution of the residuals in the prediction of a finest-scale image of grass, using one previous scale image. The analytical fit to this sample CDF is achieved with a log-Rayleigh distribution; we note that the match is so good that the two curves are difficult to distinguish. The sample CDF associated with targets summarizes the aggregate statistics of the residuals in the prediction of 136 finest-scale images of tactical targets, using two previous scale images. The analytical fit to this sample CDF is achieved with a Gaussian distribution.

line) is with a Gaussian distribution,

$$P_{w_s}(w) = \frac{\exp[-w^2/(2\sigma^2_{m(s)})]}{\sqrt{2\pi}\sigma_{m(s)}}, \tag{4.5}$$

where $\sigma_{m(s)}$ is chosen to match the sample standard deviation associated with the training data. We note that the match between measurement and analytical fit is reasonably good. Although not shown in the figure, the same reasonably good match is also obtained for coarser-scale residuals, where we continue to use a Gaussian fit.

We now consider the CDFs in Figures 4-7 and 4-8 for the residuals for our natural-clutter model. The measured sample CDF (i.e., the solid curve) is based upon the residuals in the prediction of a 256×256 finest-scale image of a homogeneous region of grass, using the 128×128 second-finest scale image. Our corresponding analytical fit (i.e., the dotted line) is with a zero-mean log-Rayleigh distribution,

$$P_{w_s}(w) = k \exp\left[kw - \gamma - \exp(kw - \gamma)\right], \tag{4.6}$$

where

$$k = \frac{\ln 10}{10}$$
$$\gamma \approx 0.57721566 \text{ (Euler's constant)}.$$

The log-Rayleigh distribution is closely related to the complex Gaussian distribution, which in turn is frequently used to characterize the statistics of speckle. To elaborate on

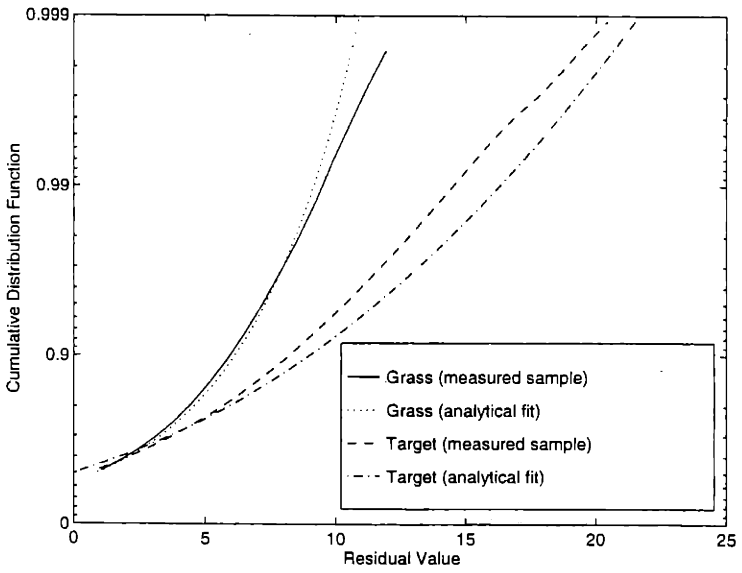


Figure 4-8: CDFs for prediction residuals associated with both our man-made model and our natural-clutter model. These distributions are identical to the ones in the previous figure; here, however, the axes have been scaled to focus exclusively on the upper tails of the distributions.

this connection, let us consider the radar reflectivity measured in a given resolution cell with a SAR sensor. This reflectivity is often modeled as $X + jY$, with X and Y are independent, identically distributed, zero-mean Gaussian random variables (and $j = \sqrt{-1}$) [53]. This speckle model has been justified, both theoretically, by appealing to the Central Limit Theorem, and experimentally, with actual radar sensor measurements. From the complex-valued random variable $X + jY$ we can obtain a log-Rayleigh random variable as $\log(\sqrt{X^2 + Y^2})$.

We note in Figure 4-7 that the match between our sample CDF for grass residuals and our analytical fit is quite good, at least up to the CDF level of 0.99. Although not shown in the figure, the match is equally good when a log-Rayleigh distribution is used to model the residuals at coarser scales.

State-space representation of models

With the model identification procedure now completed, we recast our final model choices in state-space form. This recasting provides a convenient way to summarize our model choices. Furthermore, the state-space form brings out clearly the models' Markovian properties, which will be central to the efficiency of our likelihood calculations.

The state $x(s)$ is defined to be an R-dimensional vector, containing the pixel value $I(s)$ together with the pixel values residing at the $R-1$ ancestors of node s. For our natural-clutter model, $R = 1$ and for our man-made model, $R = 2$, and thus we have

$$x(s) = \begin{cases} I(s) & \text{(natural-clutter model),} \\ \begin{pmatrix} I(s) \\ I(s\bar{\gamma}) \end{pmatrix} & \text{(man-made model).} \end{cases} \tag{4.7}$$

To be consistent with (4.3), the scale-recursive dynamics for $x(s)$ are thus defined to be

$$x(s) = \begin{cases} a_{1,m(s)}x(s\bar{\gamma}) + w(s) & \text{(natural-clutter model),} \\ \begin{pmatrix} a_{1,m(s)} & a_{2,m(s)} \\ 1 & 0 \end{pmatrix} x(s\bar{\gamma}) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} w(s) & \text{(man-made model).} \end{cases} \tag{4.8}$$

In this recursion, $w(s)$ is scalar valued and white (i.e., $w(s)$ is independent of $w(\sigma)$, for $s \neq \sigma$). For the natural-clutter model, the distribution of $w(s)$ is log-Rayleigh, given by (4.6), while for the man-made model, the distribution is Gaussian, given by (4.5). The values of the model-dependent regression coefficients $\mathbf{a}_{i,m(s)}$ are given in Table 4.2.

Since the pixel values $I(s)$ are directly observable, the measurements $y(s)$ must be noiseless. In fact, we define $y(s)$ to be

$$\begin{aligned} y(s) &= \begin{cases} x(s) & \text{(natural-clutter model),} \\ \begin{pmatrix} 1 & 0 \end{pmatrix} x(s) & \text{(man-made model).} \end{cases} \\ &= I(s). \end{aligned} \tag{4.9}$$

4.3 Description of Discrimination Algorithms

Now that we have identified stochastic models for SAR imagery of man-made objects and natural clutter, we are prepared to confront directly the problem of automatic discrimination between these two image types. We describe two discrimination algorithms, both designed for application to ROIs cued by a prescreening algorithm. The first focuses on size, texture and contrast characteristics of single-resolution imagery, and represents an optimized version of the standard discriminator used in the Lincoln Laboratory ATR system. The second represents an extension of the first, to include multiresolution characteristics in the decision-making process.

We begin by describing the procedure for calculating our so-called multiresolution discriminant. Then, we describe each of the two discrimination algorithms. We emphasize that the multiresolution discriminant is used only in the second of these.

4.3.1 Calculation of Multiresolution Discriminant

To motivate the structure of our new multiresolution discriminant, we recall that our multiscale stochastic models provide an implicit characterization of the joint statistical distribution of the pixel values in the multiresolution sequence of images $I_0, I_1, \ldots, I_L$. In particular, our models implicitly define the two conditional probability density functions (PDFs)

$$P_{I_0,I_1,\ldots,I_L|man-made}(I_0, I_1, \ldots, I_L \mid man-made) \quad \text{and}$$
$$P_{I_0,I_1,\ldots,I_L|natural-clutter}(I_0, I_1, \ldots, I_L \mid natural-clutter)$$

As we describe below, these conditional PDFs can be calculated efficiently. Thus, with their ready availability, we are led naturally to a modified Neyman-Pearson formulation of the discrimination problem. In this formulation, we seek to minimize the probability of false alarm (i.e., the probability that a natural-clutter ROI is incorrectly classified), subject to a fixed probability of detection (i.e., the probability that a man-made ROI is correctly

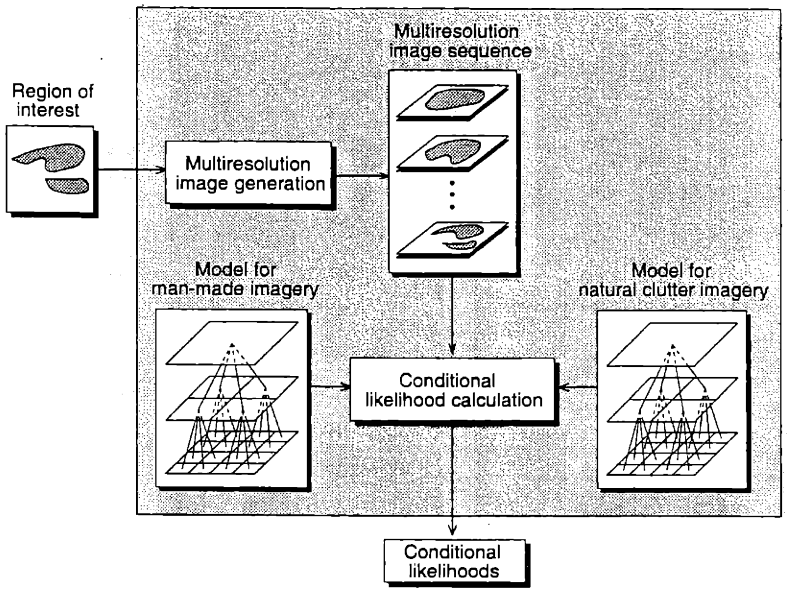


Figure 4-9: Schematic description of the calculation of our multiresolution discriminant. An input region of interest is coherently processed into a sequence of lower resolution images. Likelihoods are then evaluated, conditioned on each of the multiscale stochastic models being correct. The multiresolution discriminant is defined to be the logarithm of the ratio of these likelihoods.

classified). Classical results assure us that the optimal processor (assuming that our models represent truth) is a likelihood ratio test [63], and motivated by this fact, we define our multiresolution discriminant to be the logarithm of the likelihood ratio.

A schematic description of the procedure for calculating this discriminant is provided in Figure 4-9. As indicated in the figure, we begin by coherently processing a given ROI into a multiresolution sequence of images. Then, we perform a classical likelihood ratio calculation, evaluating the likelihood of the multiresolution sequence, conditioned on each of our models being correct.

Likelihood calculations clearly play an essential role in this procedure. Fortunately, the Markovian structure of our models leads to simple, explicit likelihood expressions that can be computed efficiently. To describe them, we first introduce some convenient notation. We let H_0 (H_1) denote the hypothesis that the ROI represents natural clutter (a man-made object). We let M_0 and M_1 denote the coarsest and finest scales, respectively, for which we have observations (i.e., $M_0 = M - L + 2$ and $M_1 = M$). We define Y to be a vector containing all the observations $y(s)$. We let $\mathbf{a}_{k,H_0}$ ($\mathbf{a}_{k,H_1}$) denote the kth scale regression coefficients for the natural-clutter model (man-made model). We let $w_{H_i}(s)$ denote the residual in the autoregressive prediction of the pixel value $I(s)$, using the model underlying H_i; in keeping with the conventions established in Section 4.2.2, $w_{H_i}(s)$ is given by

$$w_{H_i}(s) \;=\; y(s) - \mathbf{a}^T_{m(s),H_i} x(s\bar{\gamma}). \tag{4.10}$$

Finally, we define $P_{w_s|H_0}(w_s \mid H_0)$ ($P_{w_s|H_1}(w_s \mid H_1)$ to be a log-Rayleigh distribution (4.6) (Gaussian distribution (4.5)).

With these notational conventions established, the multiresolution discriminant can be

expressed in the following way:

$$\log\left(P_{Y|H_1}(Y \mid H_1)\right) \; - \; \log\left(P_{Y|H_0}(Y \mid H_0)\right). \tag{4.11}$$

In this expression, each log-likelihood term has a simple decomposition that renders it easy to compute. As justified in detail in Appendix C.2, this decomposition takes the following form:

$$\log\left[P_{Y|H_i}(Y \mid H_i)\right] \;=\; \sum_{k=M_0}^{M_1} \sum_{\{s;\, m(s)=k\}} \log\left[P_{w_s|H_i}\left(w_{H_i}(s) \mid H_i\right)\right]. \tag{4.12}$$

Each summand in this decomposition represents a penalty associated with a single residual. This penalty provides a quantitative measure of the mismatch between actual data and our models' predictive fit to these data. For example, under hypothesis H_1, we can use (4.5) to express the penalty in the following familiar quadratic form:

$$\log\left[P_{w_s|H_1}\left(w_{H_1}(s) \mid H_1\right)\right] \;=\; -\frac{w_{H_1}^2(s)}{2\sigma_{m(s)}^2} - \log\left(\sqrt{2\pi}\sigma_{m(s)}\right). \tag{4.13}$$

Combining (4.11) and (4.12), we see that for a given multiresolution sequence of images, our multiresolution discriminant can be calculated via a straightforward three-stage procedure:

1. Calculate the prediction residuals $w_{H_i}(s)$ for $i = 0, 1$.
2. Calculate the penalty $\log\left[P_{w_s|H_i}\left(w_{H_i}(s) \mid H_i\right)\right]$ associated with each residual $w_{H_i}(s)$.
3. Sum these penalties, as prescribed by (C.2) and (C.3).

4.3.2 Standard discriminator

We now describe the discriminator that is traditionally used in the second stage of the Lincoln Laboratory ATR system. The main idea underlying this discriminator is that an ROI has a number of characteristics, or so-called *features*, whose statistical distribution will significantly depend on whether the ROI represents a man-made object or natural clutter. In Table 4.3, we identify nine such features, spanning the categories *texture*, *size* and *contrast*, and measured using finest-resolution imagery only. These features comprise all the ones available to the standard Lincoln Laboratory discriminator. A brief description of each is provided in Appendix C.4, while a much more detailed description can be found in [40].

The approach used for processing features into a discrimination decision is based on a so-called one-class classification scheme [25,40]. To describe this scheme, let us suppose we have assembled a small number of measured, scalar-valued features into a vector Z. We assume that the conditional PDF $P_{Z|target}(Z \mid target)$ is known, and this PDF alone is used to make the discrimination decision, using the rule

$$\text{Declare} \left\{ \begin{array}{c} \text{target present} \\ \text{target absent} \end{array} \right\} \text{ if } P_{z|target}(Z \mid target) \left\{ \begin{array}{c} > \\ \leq \end{array} \right\} T', \tag{4.14}$$

where T' is a threshold parameter.

Category	*Feature*
Textural	standard deviation
	fractal dimension
	rank fill-ratio
Size	mass
	diameter
	rotational inertia
Contrast	peak CFAR
	mean CFAR
	percent bright CFAR

Table 4.3: This table identifies and categorizes the nine features that are available for use in the Lincoln Laboratory discriminator. A brief description of each is provided in Appendix C.4, while a much more detailed description can be found in [40].

To implement the decision rule in (4.14), we must of course specify the conditional PDF $P_{z|target}(Z \mid target)$. In this regard, an empirical analysis in [40] demonstrated that for many choices for features, the conditional distribution of Z is approximately Gaussian. Thus, the decision rule (4.14) can be written more explicitly as

$$\text{Declare} \left\{ \begin{array}{c} \text{target present} \\ \text{target absent} \end{array} \right\} \text{ if } (Z - M_t)^T \Sigma_t^{-1} (Z - M_t) \left\{ \begin{array}{c} \leq \\ > \end{array} \right\} T, \tag{4.15}$$

where T is another threshold parameter, and where M_t and Σ_t are the mean and covariance, respectively, of the Gaussian distribution. This rule is known as a *quadratic discriminator*, and is straightforward to implement, once estimates for the distribution parameters M_t and Σ_t have been computed. These parameters are estimated off line, using appropriate training imagery.

In the numerical experiments in the next section, we do not actually use the decision rule in (4.15); instead, we use an optimized version, in which a simple, but very effective modification is incorporated. To describe this modification, we first note that the diameter size feature is a powerful discriminant that has been found empirically to work best when it is used in isolation. Thus, the actual decision rule that we use consists of two stages, and works in the following way. In the first stage, the diameter feature is evaluated; only ROIs having a diameter within a prespecified range are passed to the second stage, while the others are assigned to the non-target class. In the second stage, the remaining ROIs are processed using the quadratic discriminator described in (4.15).

As a final note, we have so far bypassed a discussion of which particular features from Table 4.3 are used and how they are chosen. We will address this issue in the Section 4.4.

4.3.3 Standard discriminator with multiresolution discriminant

Our second discrimination algorithm represents extension of the first, in which the feature set used to make the discrimination decision is augmented to include the multiresolution discriminant we developed in Section 4.3.1. The structure of our decision rule here is identical to the structure of the first algorithm's decision rule: the only difference between

the two algorithms is that now our new, powerful multiresolution feature is available.

4.4 Performance of the Discrimination Algorithms

In this section, we present the detection performance results obtained by applying our discrimination algorithms to an extensive data set of actual SAR imagery. Our objective is to evaluate the detection-performance improvement that can be achieved by incorporating our new multiresolution discriminant into the standard Lincoln Laboratory discriminator. We do so by comparing the two discrimination algorithms described in Section 4.3.

4.4.1 SAR Imagery Used in Study

For our study, we have used actual imagery gathered with the Lincoln Laboratory millimeter-wave SAR. All of this imagery has 0.3-meter resolution (in both range and cross-range) and has HH polarization. There are two components to this data set:

- A training data set, used to build our two multiscale stochastic models and to estimate the parameters M_t and Σ_t associated with the conditional PDF $P_{Z|target}(Z \mid target)$
- A testing data set, used to test the discrimination algorithms.

The training data set, in turn has two components:

- A SAR image of a large, homogeneous region of grass. This is used to build our natural-clutter multiscale model.
- A collection of 136 SAR images, each representing an umcamouflaged tactical target. This collection is used both to build our man-made multiscale model, and to estimate the parameters M_t and Σ_t associated with $P_{Z|target}(Z \mid target)$.

The testing data set contains imagery representing 56 square kilometers of Stockbridge, New York. Included in the imagery are the following items:

- 136 tactical targets (i.e., 68 tanks and 68 howitzers) that are realistically deployed with radar camouflage netting,
- a large number of man-made clutter objects, including powerline towers, a farmhouse, a golf course clubhouse. and a junkyard (complete with buildings, a crane and old military jeeps), and
- natural clutter regions of trees, grass, and shrubs.

In the sequel, we refer to this testing data set as the Stockbridge imagery.

4.4.2 Generation of ROIs

To prepare for evaluation of discrimination performance, we must generate a collection of ROIs. We proceed in two steps, using the Stockbridge imagery.

In the first step, we apply the Lincoln Laboratory prescreening algorithm (i.e., the first stage of the Lincoln Laboratory ATR system) to the Stockbridge imagery. We adjust the sensitivity of this algorithm so that none of the 136 camouflaged tactical targets are discarded. At this sensitivity level, the prescreening algorithm yields 136 ROIs representing

ROI Generation step	*Tactical targets*	*Total FAs*	*Man-made FAs*	*Natural FAs*
Prescreener	136	8739	2266	6473
Size filter	136	2071	849	1222

Table 4.4: This table summarizes the outcome of our two steps for generating a collection of ROIs that can be used to evaluate discrimination performance.

tactical targets and an additional 8739 ROIs representing false alarms, both natural and man-made. Each ROI consists of 128 × 128 pixels, corresponding to a region of approximately 38 square-meters.

In the second step, we apply the size filter described in Section 4.3.2 to the ROIs generated by the prescreening algorithm. We recall that this size filter is used by both of the discriminators described in Section 4.3. The filter has the effect of eliminating all ROIs whose principal object's diameter is not within the range of diameters we expect a tactical target to have. Again, we adjust the sensitivity of this filter so that none of the 136 ROIs representing tactical targets are discarded.

Table 4.4 summarizes the outcome of the two steps we have just described, by categorizing the ROIs that are generated.

4.4.3 Standard Lincoln Laboratory Discriminator Vs. New Discriminator

We now subject the collection of remaining ROIs to a quadratic discriminator. In keeping with our objective to evaluate the effectiveness of the multiresolution discriminant, we consider two versions of the quadratic discriminator. The sole difference between the two is that each has a distinct set of features available for use in the decision process. These two sets are summarized as follows:

- The first set $\mathcal{S}_{standard}$ contains the features listed in Table 4.3; these are the ones traditionally available to the Lincoln discriminator.

- The second set $\mathcal{S}_{mr-aug}$ is an augmented version of the first set, in which the multiresolution discriminant is added.

For each version of the discriminator, we evaluate the effectiveness of every possible combination of features, where these combinations are simply subsets of the available features. In this way, for each version of the discriminator, we search for the feature combination that results in the smallest number of non-target ROIs being classified as targets, subject to the constraint that all 136 of the ROIs actually containing tactical targets are correctly classified.

The resulting optimal features, corresponding to the sets $\mathcal{S}_{standard}$ and $\mathcal{S}_{mr-aug}$, are as follows:

- $\mathcal{S}_{standard}$: standard deviation, fractal dimension, peak CFAR and percent bright CFAR,

- $\mathcal{S}_{mr-aug}$: peak CFAR, mean CFAR, new multiresolution discriminant.

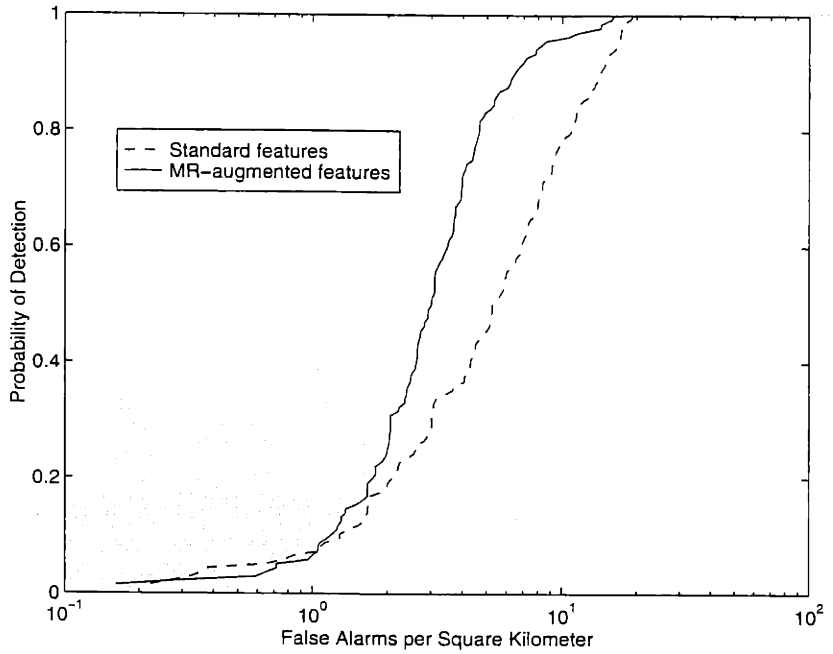


Figure 4-10: ROC curves summarizing performance of the discrimination algorithms when applied to HH-polarization SAR imagery representing 56 square-kilometers of Stockbridge, New York. The dashed and solid curves summarize, respectively, the performance of the discriminator, using the optimal combination of features from the sets $\mathcal{S}_{standard}$ and $\mathcal{S}_{mr-aug}$. We note that the performance of our new discriminator, corresponding to $\mathcal{S}_{mr-aug}$, represents a substantial and statistically significant improvement over the standard Lincoln discriminator.

	Natural FAs		Man-made FAs		Total FAs	
P_D	$S_{standard}$	S_{mr-aug}	$S_{standard}$	S_{mr-aug}	$S_{standard}$	S_{mr-aug}
0.8	52	5	563	256	615	261
0.9	117	8	676	350	793	358
0.95	191	34	753	429	944	463
1.0[a]	300	241	786	667	1086	908

Table 4.5: This table summarizes the discrimination performance at four particular operating points of the ROC curves in Figure 4-10. The first column identifies the probability of detection P_D at the operating points of interest. The next three pairs of columns list the corresponding number of false alarms generated by the discrimination algorithms.

[a] As a caveat, the results at this operating point are not statistically significant; for virtually all practical, non-degenerate problems, the operating point $P_D = 1$ implies that $P_{FA} = 1$.

The corresponding discrimination results are summarized by the receiver operating characteristic (ROC) curves shown in Figure 4-10 and in the Table 4.5.

These results merit a number of comments. First, we note that the performance of our new discriminator, using the optimal features in the set $\mathcal{S}_{mr-aug}$, represents a substantial and statistically significant improvement over the standard Lincoln discriminator. Consider, for example, the operating point $P_D = 0.95$: the new discriminator reduces the number of natural-clutter false alarms by almost a factor of six.

Second, we note that the optimal feature combination corresponding to $\mathcal{S}_{mr-aug}$ is in a certain sense consistent with the optimal feature combination corresponding to $\mathcal{S}_{standard}$. To clarify this comment, we recall that the features *standard deviation* and *fractal dimension* are part of the optimal combination corresponding to $\mathcal{S}_{standard}$. But our new multiresolution discriminant essentially captures both of these characteristics: standard deviation information is directly captured in the structure of the log-likelihood ratio, as expressed in (4.12), and fractal characteristics are fundamental to the structure of our multiresolution models and processes, as discussed in [9]. This observation is reinforced by the features in the optimal combination corresponding to $\mathcal{S}_{mr-aug}$, where use of the new multiresolution discriminant essentially supersedes joint use of *standard deviation* and *fractal dimension.*

Finally, we emphasize that our multiscale model are extremely simple, and we would expect to obtain even better performance if we used more sophisticated models. For example, as we saw in Section 4.2.2, our man-made model does not completely capture the scale-to-scale statistical coupling of a multiresolution sequence of images of a man-made object. The discrimination results obtained here certainly provide motivation for future work on developing more sophisticated models.

4.5 Conclusion

We have developed and extensively tested a new algorithm for discriminating man-made objects from natural clutter in SAR imagery. This algorithm has been extremely successful, as it has exploited the characteristically distinct variations in speckle pattern for imagery of man-made objects and of natural clutter, as image resolution is varied from coarse to fine.

Within our multiresolution framework, we used actual SAR imagery to identify a pair of multiscale models: one for SAR imagery of natural clutter and another for imagery of man-made objects. We then used these models to define a multiresolution discriminant as the likelihood ratio for distinguishing between the two image types, given a multiresolution sequence of images of an ROI. We incorporated our new discriminant into an existing, established discriminator that was developed at Lincoln Laboratory as part of a complete ATR system. To classify a given ROI, we merged the information provided by our new discriminant with the measured values of a small number of size and brightness features. We applied the resulting, new discriminator to an extensive data set of 0.3-meter resolution, HH polarization imagery. The detection results were impressive. In particular, we demonstrated a substantial and statistically significant improvement in the receiver operating characteristics when we augmented Lincoln's standard discriminator with the new discriminant. This result is surprisingly good, in light of the number of years over which the standard discriminator has been developed and refined; the result conclusively demonstrates that multiresolution methods have an effective and important role to play in SAR ATR algorithms.

Chapter 5

An Overlapping-Tree Approach to Modeling and Estimation

5.1 Introduction

In spite of the success of the multiscale approach to estimation with regard to computational efficiency, mean-square estimation error, and ability to supply error covariance information, the approach, as developed up to this point in time, has a characteristic that would appear to limit its utility in certain applications. Specifically, estimates based on the types of multiscale models described so far tend to exhibit a visually distracting blockiness [46]. Actually, we saw this blockiness firsthand in Section 3.6.3 in the context of multiscale representations of isotropic random fields.

While various interpretations of and ways to overcome this blockiness have been developed, discussed, and shown to be more than adequate in particular applications, none of these offers a completely satisfactory resolution of this issue in general. As an example of this discussion and interpretation, the authors in [46] argue correctly that in many applications, the construction of fine-scale estimates is not supported by the quality of available data, and in such cases, only coarser scale estimates are statistically significant. In these applications, one should be suspicious of *any* fine-scale estimate of the field in question, and any corresponding blockiness has a complete lack of statistical significance. However, in other applications, such as the problem of estimation of the ocean surface height [22] or the investigation of surface reconstruction in [24], multiscale-based estimates are subsequently used in a manner that requires the calculation of surface gradients; in these cases, there is an essential need for having smooth estimates, so that the gradients can be calculated meaningfully.

Although estimate blockiness can be eliminated by simple post-processing (e.g., the application of a low pass filter), the resulting increase in smoothness and visual appeal comes at a price. In particular, the post-processing can render less clear the proper interpretation of error covariance information provided by the estimation algorithm, and it limits the resolution of fine-scale details in the post-processed estimate, since the added smoothness is achieved by spatial blurring. As an alternative, our work in Chapter 3, and the work in [43,45] has demonstrated that multiscale models can be constructed that produce arbitrarily accurate representations of broad classes of random fields, including those with considerable smoothness. However, in order to achieve a high level of smoothness, the methods described in Chapter 3 and in [43,45] by themselves require the use of multiscale processes in which the

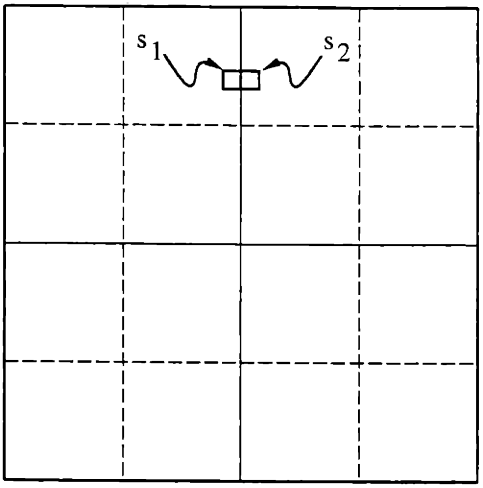


Figure 5-1: Two nodes, s_1 and s_2 that are close neighbors in physical space, but are distantly separated in tree space.

state vectors $x(s)$ have fairly high dimension, thereby leading to a sacrifice of the significant computational advantages that the multiscale modeling framework offers.

The preceding remarks suggest that for applications in which the computational efficiency of the multiscale framework is desired, but where blockiness is unacceptable, we have considerable motivation for seeking a new approach to both multiscale modeling and estimation. In this chapter, [1] we consider a novel approach that yields the desired effect. Our approach simultaneously achieves three objectives:

1. It yields low-dimensional multiscale models that are quite faithful to prespecified random field covariance structures to be realized, and thus admits an extremely efficient, optimal (or nearly optimal) estimation algorithm;

2. The resulting estimation algorithm retains one of the most important advantages of the multiscale estimation framework, namely the efficient computation of estimation error covariances;

3. Both the multiscale models and the corresponding estimation algorithm eliminate the blockiness associated with previously developed multiscale models and estimates.

In contrast to standard multiscale processing [23, 46], which achieves objectives one and two, and to standard multiscale processing with simple post-processing [46], which achieves objective one and partially achieves objective three, our approach accomplishes all three objectives.

5.2 Overview of Approach

To describe our modified approach to multiscale modeling and estimation, let us begin with a more careful look at the source of blockiness that is typical of estimates produced by the standard multiscale approach. Towards this end, we consider a multiscale process indexed

[1] The work in this chapter was done in collaboration with fellow graduate student Paul Fieguth.

on a q-th order tree. As we know from Chapters 2 and 3, the state $x(s)$ at any given node of such a tree represents an appropriate, aggregate description of the subset of the finest-scale process that descends from the given node. More specifically, the designated role of the state $x(s)$ in a multiscale process is to store enough information to decorrelate the values of the process in the corresponding $q+1$ subtrees of nodes extending away from the given node s. This decorrelation role is what leads both to efficient estimation algorithms and to the source of the blockiness problem.

We can clearly see the connection between the decorrelating role of state information and the blockiness problem by considering Figures 1-1 and 5-1. Focusing on the upper-left and upper-right quadrants of the image domain depicted in Figure 1-1, we note that these two quadrants are separated at the coarsest level of the tree, and therefore all of the correlation between any two finer scale pixels in the two quadrants, such as s_1 and s_2 in Figure 5-1, must be completely captured in their common ancestor, namely the root node s_0 at the coarsest scale of the tree. In this sense, the pixels s_1 and s_2 may be close physically, but they are separated considerably in terms of the distance to their nearest common ancestor node. We refer to this latter distance as so-called *tree-distance*; with respect to tree distance, pixels s_1 and s_2 are far apart. High local correlation between such spatially close neighbors, as one might expect if the field being modeled has some level of regularity or smoothness, translates into $x(s_0)$ having a high dimension, in essence to keep track of all of the correlations across quadrant boundaries.

One way to reduce this high dimensionality is to identify and retain only the principal sources of correlation across boundaries at each level on the tree. Keeping only these principal sources effectively achieves maximal decorrelation of descendants with minimal dimension of state variables. Indeed, Chapter 3 was devoted to developing a systematic procedure for identifying the needed principal sources of decorrelating information, and to building multiscale models of any desired fidelity. While this approach by itself can yield low-dimensional models of sufficient fidelity for many applications (such as texture discrimination [44] or problems such as that in [46] where only coarse-scale estimation is meaningful), it cannot overcome the blockiness problem. In particular, neglecting even a small amount of correlation at a coarse level of the tree can cause noticeable irregularities across boundaries such as that separating s_1 and s_2 in Figure 5-1, and thus an additional element must be introduced.

In this chapter, we introduce the needed additional element by discarding the standard assumption that distinct nodes at a given level of our tree correspond to disjoint portions of the image domain. Instead we construct models in which distinct tree nodes correspond to *overlapping* portions of the image domain. As a consequence of this simple idea, which was first used in [22,24], a given image pixel at, say, the finest scale may now correspond to several tree nodes at this finest scale. In this way, we remove the hard boundaries between image-domain pixels such as s_1 and s_2 in Figure 5-1. These hard boundaries are eliminated, because now multiple tree nodes contribute to each of these pixels, thus reducing the tree distance between the nodes corresponding to these pixels and spreading the correlation that must be captured among a set of nodes. For obvious reasons, we refer to these multiscale models as overlapping-tree models.

We use these overlapping-tree models for both modeling and estimation, as depicted abstractly in Figure 5-2. In both of these contexts, we start with assumed knowledge of the correlation structure P_χ of some random field χ. Corresponding to this random field χ, we devise a so-called *lifted-domain* version χ_l, where this lifted-domain field lives at the finest-scale of an overlapping-tree multiscale representation of χ. The mapping from χ to

χ_l is denoted by $\chi_l = G_x\chi$, where we emphasize that this operator G_x is one-to-many: the lifted-domain field χ_l has more pixels than the image-domain field χ. To map back from χ_l to χ, we devise an operator H_x having two important properties: (i) the field $H_x\chi_l$ has exactly, or nearly exactly, the same correlation structure as χ; (ii) the field $H_x\chi_l$ is guaranteed to have the desired level of smoothness.

In the top half of Figure 5-2, we depict an application of our overlapping-tree models to the problem of efficiently generating sample paths of a random field having the prespecified correlation structure P_χ. Given P_χ, a low-order multiscale model is built to approximately realize the correlation structure of the overlapped field χ_l; we denote this correlation by P_{χ_l}, where $P_{\chi_l} = G_x P_\chi G_x^T$. Because of the low order of this multiscale model, sample paths can be generated in a computationally efficient manner, and by post-processing these sample paths with the smoothing operator H_x, we obtain sample paths of a random field that are guaranteed to be smooth and that approximately have the desired correlation P_χ. We have already essentially addressed in Chapter 3 the technical problem of constructing the tree model. Here we address the additional issue of devising the lifting and interpolation operators G_x and H_x, and we combine these operators with the model-building tools of Chapter 3 to meet the three objectives of low-dimensional states on the tree, accurate approximation of the desired second-order statistics of the field χ, and the generation of fields without blocky artifacts.

In the bottom half of Figure 5-2, we depict an application of our overlapping-tree models to the problem of optimal estimation of the value of a random field χ, given noisy observations y. For this purpose, we devise an operator G_y that plays a role directly analogous to the role of G_x: the operator G_y lifts the actual observations y of the random field, to yield lifted-domain observations y_l of the random field χ_l. These observations are then processed by our efficient multiscale tree algorithm to produce an estimate $\hat{\chi}_l$ which is then projected back to yield $\hat{\chi}$, the desired estimate of the random field. The low dimensionality of the multiscale model allows the estimation calculations to be carried out in an extremely efficient manner, and the properties of the operator H_x guarantee that the field estimates can be generated to have the desired level of smoothness. We address the technical problem of justifying the optimality, or near optimality of this estimation procedure, and we also demonstrate that estimation error covariance information can be generated in an efficient and meaningful way.

This chapter is organized in the following way. After introducing some convenient, special notation, we introduce all the components of our approach to modeling and estimation, including a more complete description of the operators G_x, G_y, and H_x. In this fashion, we identify the precise technical challenges to be confronted to develop fully the approach. We then characterize the optimality properties of our estimation procedure. Next, we develop an implicit scheme for describing the projection operators to and from the overlapped domain, and finally we illustrate the effectiveness of our new approach to modeling and estimation by means of five examples.

5.3 The Estimation Operator

For the purposes of our development in later sections of this chapter, it will be useful to have an explicit input-output expression for the optimal linear least-squares estimator. Specifically, suppose that we wish to estimate a random vector χ based on linear observations

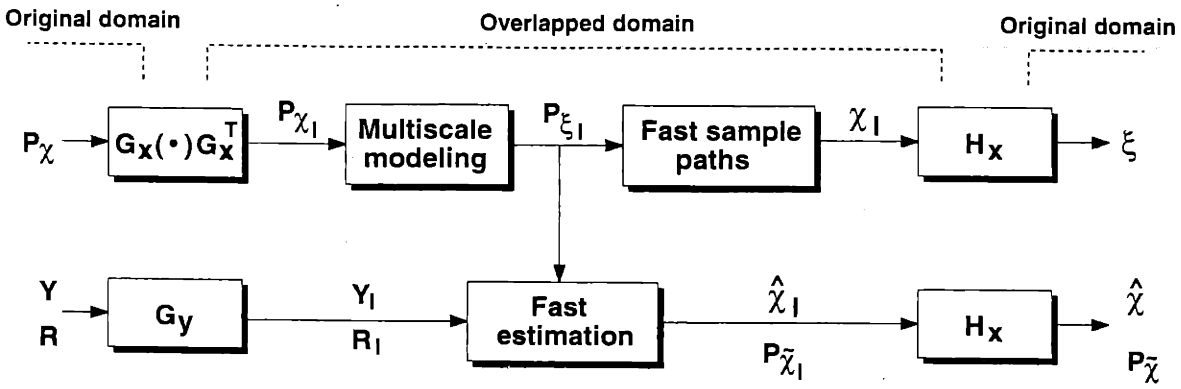


Figure 5-2: An abstract view of our overlapping-tree approach to multiscale-based modeling and least-squares estimation. Fast multiscale estimation and sample-path generation (producing possibly blocky ξ_l and $\hat{\chi}_l$ respectively) are accomplished in the overlapped domain. G_x projects the statistics of χ into the overlapped domain; G_y projects measurements y into the domain; and H_x, which possesses certain smoothness properties, projects the estimates $\hat{\chi}_l$ back out of the overlapped domain.

y, where

$$y = C\chi + v. \tag{5.1}$$

Here v denotes the measurement noise or error, assumed to be zero mean and covariance R and to be uncorrelated with χ. For simplicity, we assume that χ is zero mean[2] with covariance P_χ. Then, a standard result in linear least-squares estimation is that the optimal estimator can be expressed in input-output form as

$$\hat{\chi} = Ly \tag{5.2}$$

where[3]

$$L = P_\chi C^T (CP_\chi C^T + R)^{-1} \tag{5.3}$$

and the associated error covariance is given by

$$P_{\tilde{\chi}} = P_\chi - P_\chi C^T (CP_\chi C^T + R)^{-1} CP_\chi = P_\chi - LCP_\chi. \tag{5.4}$$

While the multiscale estimation algorithm described Section 2.1.5 calculates $\hat{x}(s)$ and the error variance $P_{\tilde{x}(s)}$ in a recursive manner, taking advantage of the Markov structure of multiscale processes (rather than by explicit matrix-vector multiplication, as in (5.2)), it will be useful in subsequent sections to have such an input-output view available. Furthermore, although our multiscale framework applies to the general case, we focus in this chapter exclusively on a particular case that, as we will see, corresponds to the problem of estimating a scalar random field (e.g., an image), given noisy (and possibly sparse) point measurements of the field. Specifically, we assume all attention focuses on the finest scale, so that observations are only available at that scale and only the fine-scale estimates are of interest.

[2]Otherwise we can subtract out its mean m_x first and simply add it back after estimation of $(x - m_x)$.

[3]Obviously for these expressions to make sense, the indicated inverses must exist. While there is no conceptual difficulty in extending these ideas to the singular case in which R is *not* invertible, for simplicity, we assume throughout our development that R *is* invertible.

Furthermore, we assume that at this finest scale, both the state and the measurements are scalar valued.

If $s_1, s_2, \ldots, s_N$ denote the nodes of the tree at the finest scale, then we can identify the vector χ to be estimated as the vector of the $x(s_i)$ ordered sequentially. Similarly, we let $s_{i_1}, s_{i_2}, \ldots, s_{i_M}$ denote the subset of these nodes where we have measurements,

$$y(s_{i_j}) = C(s_{i_j})x(s_{i_j}) + v(s_{i_j}) \tag{5.5}$$

where the measurement noises $v(s_{i_j})$ are zero mean, uncorrelated with each other, and have variances $R(s_{i_j}) \neq 0$ (so that there are no measurements that are perfect). We let y denote the vector of the measurements $y(s_{i_j})$, ordered sequentially, and we define v analogously in terms of the noise terms $v(s_{i_j})$. Then, y, χ and v are related as in (5.1), where C is a matrix determined by (5.5) and the construction of y, and v has covariance $R = \mathrm{diag}(R(s_{i_1}), R(s_{i_2}), \ldots, R(s_{i_M}))$.

5.4 Formulation of the Problems of Modeling and Estimation with Overlapping Trees

In this section, we identify the central components of our new approach to multiscale modeling and estimation with overlapping trees. As previewed in Section 5.1, three of these components include the lifting operators G_x and G_y, as well as the interpolation, or smoothing, operator H_x. There are two others, namely the lifted-domain observation matrix C_l and the lifted domain observation-noise covariance R_l, that play roles in the overlapped domain directly analogous to the roles of the matrices C and R introduced in Section 5.3. We will see that these matrices all have a great deal of structure and that there are important, simple relationships among them. Furthermore, we will see that *any* sub-optimality in our approach to estimation can be *completely* traced to our use of an approximate model to realize the correlation structure of the overlapped field χ_l; in the last part of this section, we formalize this fact with a precise statement concerning the optimality of our estimation algorithm.

5.4.1 Modeling of Random Fields with Overlapping Tree Processes

Let χ be a zero-mean random field written for simplicity as a vector, and having covariance P_χ. We now consider the problem of simulating χ, or a close approximation thereof. That is, we consider the problem of generating sample functions of a zero-mean random field with covariance equal to P_χ or close enough to P_χ so that its significant statistical characteristics are captured.

From a computational point of view, this simulation problem poses nontrivial challenges, and has been the focus of a considerable amount of research in the signal and image processing community. One notable case in which computationally efficient techniques do exist is for generation of sample functions of stationary random fields, defined on regularly sampled toroidal lattices. In this case, the 2-D FFT can be used to diagonalize the field's covariance matrix; sample functions can then be generated as discussed in detail in Section 2.2.3. However, for most other types of fields, the generation of sample paths is quite complex computationally. For example, one approach involves the following three step procedure: (i) compute the square root $P_\chi^{1/2}$ of the covariance matrix, (ii) generate a vector w of unit

variance uncorrelated random variables, and (iii) compute the sample path as $\chi = P_\chi^{1/2} w$. While this approach is conceptually straightforward, there is a considerable challenge in computing the matrix square root $P_\chi^{1/2}$, requiring in general $\mathcal{O}(K^3)$ calculations for a random field of K points. Similar computational difficulties are encountered with iterative generation methods, such as those for Markov random fields, which can frequently take an exorbitant number of iterations, especially to capture significant large-scale correlations.

On the other hand, as discussed in Section 2.1.6, the simulation of a random field having a multiscale model is extremely fast; in fact, even in the one case where fast FFT-based methods can be used, namely for generation of stationary random fields on regular lattices, the $\mathcal{O}(K)$ complexity of our method is asymptotically better than the $\mathcal{O}(K \log K)$ complexity of these FFT-based approaches. Thus, we are led to consider more completely the issues involved in an overlapping-tree approach to simulation.

An overlapping-tree approach to simulation

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

$$\chi_l = G_x \chi. \tag{5.6}$$

This lifted-domain field χ_l corresponds to the finest scale of an overlapping tree process, and acts as a particular, redundant representation of χ, having more pixels than the original field. The matrix G_x is not chosen arbitrarily; it has a considerable amount of sparse structure, as we discuss in greater detail in Sections 5.4.1 and 5.5. Furthermore, G_x is chosen such that it has a left inverse H_x,

$$H_x G_x = I, \tag{5.7}$$

satisfying certain smoothness properties to be discussed shortly.

In the second step, we combine our knowledge of the covariance P_χ and matrix G_x, together with the stochastic realization method developed in Chapter 3 to build a low-dimensional multiscale model whose finest-scale statistics are an accurate approximation to the statistics of χ_l. Specifically, from (5.6), we see that the covariance of χ_l is given by

$$P_{\chi_l} = G_x P_\chi G_x^T. \tag{5.8}$$

Then, the covariance P_{ξ_l} of ξ_l, the random field living at the finest scale of the multiscale model that we construct, satisfies

$$P_{\xi_l} \approx P_{\chi_l} \tag{5.9}$$

To generate a sample function of a random field ξ qualitatively similar to χ, we then apply the operator H_x to ξ_l:

$$\xi = H_x \xi_l. \tag{5.10}$$

This random field ξ is guaranteed to be smooth, by the assumed smoothness properties imposed on H_x. Also, thanks to (5.7)-(5.10), ξ will have approximately the same statistics as χ.

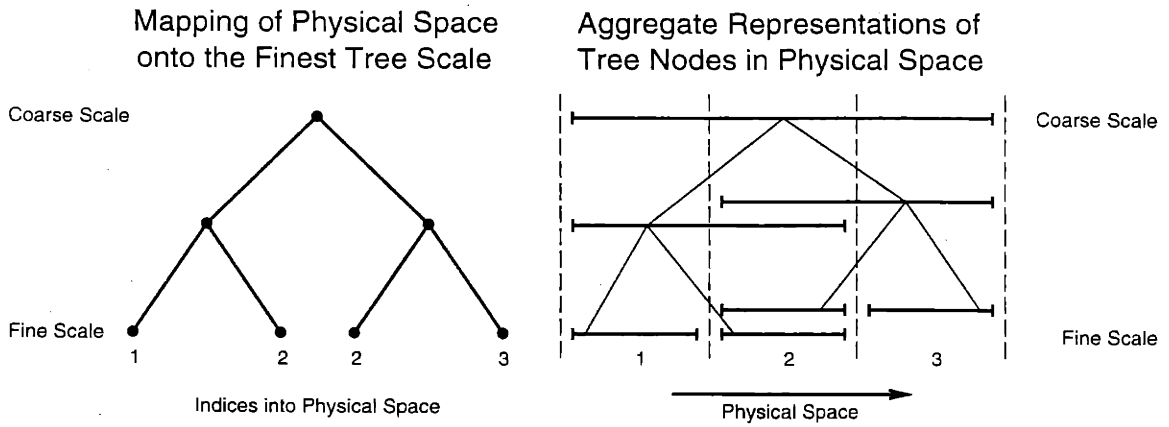


Figure 5-3: Illustration of an overlapping-tree representation of a process of length three, showing both the dyadic tree (left) on which the representation is based, and depiction (right) of the representation of each tree node. The bar $\vdash$ associated with each tree node represents the subset of the points $\{1,2,3\}$ associated with that node.

Designing G_x and H_x

Thus, the design problem confronting us is that of specifying the operators G_x and H_x and then constructing the multiscale model for ξ_l, so that the following properties hold: (i) G_x and H_x are sparse and local, (ii) H_x achieves the desired smoothness, (iii) the multiscale model is of sufficiently low dimension that simulation can be done efficiently, and (iv) the approximation (5.9) is sufficiently accurate so as to lead to sample functions with the desired characteristics captured in P_χ. The method we use here to construct the multiscale models is the canonical-correlations-based stochastic realization method described in Chapter 3. The focus of attention in the remainder of this and the next section is on the design of G_x and H_x. In Section 5.6, we then demonstrate that our approach does indeed achieve objectives (i)-(iv).

Example

To introduce the basic issues involved in specifying G_x and H_x, let us consider a very simple 1-D example of a random process of length 3. Collecting the process values into a vector $\chi^T = (\chi_1, \chi_2, \chi_3)^T$, we suppose that the covariance of χ is as follows:

$$E\left(\chi\chi^T\right) = P_\chi \equiv \begin{pmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{pmatrix} \tag{5.11}$$

Our objective is to develop an overlapping tree model for χ, indexed on a dyadic tree having four finest-scale nodes, thereby providing only a minimal amount of redundancy. Figure 5-3 displays an example of such a tree. On the right, we depict the tree with an indication of the subsets of real, physical points (i.e., subsets of $\{1,2,3\}$) to which each node corresponds. Thus, the top node corresponds to all three points (i.e., $\{1,2,3\}$) and the two nodes at the second level correspond to $\{1,2\}$ and $\{2,3\}$ respectively. At the bottom level there is a single node corresponding to data point 1 and another for 3, but there are *two* nodes corresponding to 2. That is, in the lifted domain on the tree, signal point 2 is lifted to have two finest-scale tree nodes. Thus if we order the four fine-scale nodes from left to

right and we view our lifting process as simply copying the value of signal point 2 to both of the tree nodes to which it corresponds, then we are led to the following definition of G_x:

$$G_x \equiv \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{5.12}$$

which implies that

$$P_{\chi_l} \equiv G_x P_\chi G_x^T = \begin{bmatrix} 1 & 0.5 & 0.5 & 0 \\ 0.5 & 1 & 1 & 0.5 \\ 0.5 & 1 & 1 & 0.5 \\ 0 & 0.5 & 0.5 & 1 \end{bmatrix} \tag{5.13}$$

Basic constraints on G_x and H_x

The preceding example illustrates the basic constraints that we place on any *lifting* matrix G_x:

1. It consists entirely of zeros and ones.
2. Each column has at least one nonzero entry.
3. Each row has exactly one nonzero entry.

These conditions ensure the following basic properties:

1. Every position in the original domain corresponds to at least one position in the overlapped domain.
2. Every position in the overlapped domain corresponds to exactly one position in the original domain

Thus, the lifting process is local (in that each tree node corresponds to a single point in the original domain) and is in fact the product $G_x\chi$ is trivial to compute, once G_x has been specified. This specification of G_x can be associated naturally with the overlapping structure as illustrated in Figure 5-3 or, more specifically, with the association of fine-scale nodes with pixels. Thus, depending on how one chooses an overlapping structure, a different lifting operator will generally result.

Turning to the smoothing operator, we note that the inverse relation (5.7) between H_x and G_x, together with our imposed constraints on the structure of G_x, lead to an important constraint on the structure of H_x. In particular, the value at any given point in the original-domain is equal to the weighted sum of the values of the finest-scale nodes corresponding to the given data point, where these weights must sum to unity. For example, with G_x as in (5.12), the possible choices for H_x are of the form

$$H_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & a & b & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5.14}$$

where $a + b = 1$. Here a and b can be thought of as the weights being placed on the value of the two nodes corresponding to data point 2 in order to specify χ_2. For example, equal

weighting $a = b = 1/2$ would intuitively lead to the most smoothness in the correlation structure from χ_1 through χ_3. On the other hand, it is important to emphasize that the averaging implied by (5.14) is *not* at all the same as spatial averaging, since we average only those tree points corresponding to the *same* point in real space.

5.4.2 Estimation of Random Fields with Overlapping Tree Processes

Let us turn now to the problem depicted in the bottom half of Figure 5-2. The objective is to exploit the efficiency of the multiscale estimation algorithm to perform optimal or near-optimal estimation of a random field χ, while avoiding blocky artifacts.

Suppose that we have noisy measurements of χ

$$y = C\chi + v \qquad v \sim \mathcal{N}(0, R)$$

where two conditions hold: (i) each component of y represents a measurement of an individual pixel, so that each row of C has only one nonzero entry, and (ii) the measurement noise terms are uncorrelated with each other, so that the covariance R of v is diagonal. From Section 5.3 we know that $\hat{\chi} = Ly$, where L is given by (5.3), assuming that χ has prior covariance P_χ. However for a K-pixel field the calculation of L is generally $\mathcal{O}(K^3)$ and the calculation of the product Ly is $\mathcal{O}(KN_{meas})$ where N_{meas} is the number of measurements. Virtually the only case in which this computational load can be reduced to a practical level is when the field χ is stationary, and we have dense, regularly sampled measurements of identical quality (implying that C and R are both multiples of the identity); in this special case FFT methods can reduce the computational load to $\mathcal{O}(KlogK)$. However in other cases, the $\mathcal{O}(K^3)$ computational load cannot be reduced, and in these cases, the traditional approach is to turn to iterative methods for the computation of $\hat{\chi}$. The problem here is that not only can these iterative methods be slow, but they also do not yield error covariance information.

We are thus motivated to consider the estimation approach illustrated in the bottom half of Figure 5-2. To develop this approach, we will need all the results from our approach to modeling, plus a bit more. In particular, we will need the lifting and projection operators G_x and H_x for our random field, as well as a multiscale model for ξ_l, such that $H_x\xi_l$ is an adequate approximation of the field χ; the issues related to determining these were discussed in the preceding section.[4] In addition, specific to the estimation problem, we need to define a lifting operator G_y for the measurements:

$$y_l = G_y y \tag{5.15}$$

and a lifted measurement model

$$y_l = C_l\chi_l + v_l. \tag{5.16}$$

Once these quantities have been specified, we can carry out estimation as a two-step procedure: (i) application of our multiscale estimation algorithm to estimate χ_l based on y_l, and (ii) application of H_x to the resulting estimate, thereby yielding a near-optimal estimate of χ based on y.

For step (i) to be feasible, the components of y_l must represent observations of individual

[4]The only exception is the construction of the multiscale model for ξ_l, which was described in Chapter 3.

fine-scale tree nodes, where the observation noises are uncorrelated. In other words, each row of C_l must have only one non-zero entry and the measurement covariance R_l of v_l must be diagonal. Turning then to step (ii), we can clearly see the requirements for its success by writing the multiscale estimator in input-output form as in (5.2):

$$\hat{\chi}_l = L_l y_l = P_{\chi_l} C_l^T \left(C_l P_{\chi_l} C_l^T + R_l \right)^{-1} y_l \tag{5.17}$$

Combining (5.17) with (5.15), we see that our step-(ii) objective of satisfying $\hat{\chi} \approx H_x \hat{\chi}_l$ is equivalent to satisfying

$$P_\chi C^T \left(C P_\chi C^T + R \right)^{-1} = L \approx H_x L_l G_y = H_x P_{\chi_l} C_l^T \left(C_l P_{\chi_l} C_l^T + R_l \right)^{-1} \tag{5.18}$$

Designing G_y, C_l and R_l

Assuming that G_x, H_x, and the multiscale model (which specifies P_{χ_l}) have been chosen, the remaining quantities to be specified include G_y, C_l, and R_l. Given the construction of G_x, in which each pixel is associated with a set of fine-scale nodes, the most natural choice for C_l is specified by requiring that if a real measurement is made at a particular pixel, then lifted measurements should be specified at each of the fine-scale tree nodes corresponding to that pixel. For example, for the three-point process and dyadic tree lifting illustrated in Figure 5-3, let us suppose that we have measurements of χ_1 and χ_2, namely

$$y \equiv \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad C \equiv \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} \quad R \equiv \begin{pmatrix} 3 & 0 \\ 0 & 4 \end{pmatrix} \tag{5.19}$$

Then, in our lifted domain we should have *three* measurements, one corresponding to the single node associated with χ_1, and *two* corresponding to the nodes associated with χ_2. That is,

$$C_l \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix} \tag{5.20}$$

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

$$G_y \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \qquad y_l = G_y y \equiv \begin{bmatrix} y_1 \\ y_2 \\ y_2 \end{bmatrix} \tag{5.21}$$

At first glance, this procedure appears to create a significant problem: for the multiscale estimation algorithm to work, we require that the measurements at distinct nodes have uncorrelated errors. With y_l and C_l defined as in (5.21) and (5.20) this uncorrelatedness certainly does not hold, since two of the "measurements" are identical. Nevertheless, there is no intrinsic mathematical difficulty with simply *modeling* these two measurements as being distinct ones, each of the state at the corresponding node, with uncorrelated measurement errors; indeed this modeling approach is how we proceed. However, by proceeding this

way, we appear to have created another difficulty. Specifically, by modeling y_l in this way, we appear to be asserting that we have more information than we actually do; we now have two independent measurements of the nodes corresponding to x_2. To compensate for this fact, we need to ensure that the total information in these two measurements is the same as in the single actual measurement. We make this idea precise by defining the information content of a scalar-valued measurement to be equal to the reciprocal of the measurement's noise variance; for instance, the information provided by y_2, as defined by (5.19), is 1/4. Furthermore, we define the information content of a whole collection of scalar-valued measurements of a single point (assuming uncorrelated noise terms)to be equal to the sum of the information contents of the individual measurements. In terms of these conventions, it is straightforward to ensure that the amount of information in y_l is the same as the amount of information in y; specifically, given R in (5.19), we define

$$R_l \equiv \begin{pmatrix} 3 & 0 & 0 \\ 0 & \rho_1 & 0 \\ 0 & 0 & \rho_2 \end{pmatrix}, \tag{5.22}$$

where the positive scalars ρ_1 and ρ_2 are constrained to satisfy

$$\frac{1}{\rho_1} + \frac{1}{\rho_2} = \frac{1}{4}.$$

For instance, one possibility is to let

$$\rho_1 = \rho_2 = 8.$$

5.4.3 Optimal Estimation Through Lifting and Projection

Let us now turn to a general analysis of the optimality of our overlapping-tree approach to multiscale estimation. We demonstrate that if our multiscale model for χ_l is such that the approximate equality in (5.9) is in fact an exact equality, then there exist values for the matrices G_x, G_y, H_x, C_l, and R_l so that the resulting estimate $\hat{\chi} = H_x\hat{\chi}_l$ is *exactly* equal to the optimal estimate (5.2) of χ based on y. This optimality will hold for all least-squares problems of the form

$$\begin{array}{ll} \text{Estimate} & \chi \\ \text{Given} & y = C\chi + v \end{array}$$

so long as C is a weighted selection matrix (i.e., each row of C has exactly one nonzero entry and each column has at most one nonzero entry.[5]), and the covariance of v, which we denote by R, is diagonal. The example given in Section 5.4.2 illustrated one very simple way to choose values for the parameters G_x, G_y, H_x, C_l and R_l; we show here that in general there is actually considerable flexibility in their choice.

The optimality properties of our estimation procedure are significant. They imply that any actual sub-optimality is traceable directly and completely to approximations made in

[5]These conditions are equivalent to saying that each measurement is of a distinct pixel and any pixel has at most one measurement associated with it. The latter assumption is for simplicity only; if there are multiple measurements of a single pixel, then since R is diagonal we can replace these by a single measurement obtained as the weighted average of the redundant measurements.

building a low-dimensional model for χ_l. We thus have explicit control of the complexity-accuracy tradeoff, and in Section 5.6, we provide some illustrations of how we manage this tradeoff.

As in our simple example in Section 5.4.2, we restrict ourselves to choices of G_x such that three properties hold: (i) G_x consists entirely of zeros and ones, (ii) each column of G_x has at least one nonzero entry, and (iii) each row of G_x has exactly one nonzero entry. We then must choose H_x so that $H_x G_x = I$, and while this requirement does indeed constrain H_x, it does leave some remaining degrees of freedom. In particular, as we have seen, the choice of G_x is directly related to the overlapping structure that we have chosen, which in turn specifies which fine-scale tree nodes correspond to which real pixels, and H_x then performs a weighted averaging among each set of tree nodes that correspond to each individual pixel, where there is flexibility in the choice of these weights. Thus, there is considerable freedom in the choices for G_x and H_x. Furthermore, the resulting matrices are quite sparse. On the other hand, for 2-D problems of practical interest, these matrices will be quite large, and thus any structure that can be imposed or discerned about the sparsity in G_x and H_x will be of considerable benefit. In Section 5.5 we describe how these matrices can be specified in an implicit manner that achieves a considerable reduction in storage requirements and increase in computational speed.

We now turn our attention to devising G_y and C_l. Towards this end, we note our actual measurements are $y = C\chi + v$ while our lifted measurements are computed as $y_l = G_y y$ and modeled as $y_l = C_l \chi_l + v_l$, where $\chi_l = G_x \chi$. Thus we have two expressions for how the real random field χ affects the lifted measurements y_l, namely $C_l G_x \chi$ and $G_y C \chi$. A logical requirement on C_l and G_y then is to require these two expressions to be equal for any χ:

$$C_l G_x \;=\; G_y C \tag{5.23}$$

Thus, once the value of either C_l or G_y is determined, the value of the other is automatically determined.

We now construct an appropriate matrix for C_l exactly as we did for our example. Specifically, we assume that for each real pixel measurement, we have an analogous measurement for *each* of the tree nodes corresponding to that real pixel. Thus if the jth component of y is $y_j = \alpha_j \chi_i + \text{noise}$ (where χ_i is a component of χ) then y_l will have measurements of the form

$$(y_l)_n = \alpha_j \, (\chi_l)_n + \text{noise} \tag{5.24}$$

for each n such that finest-scale node $(\chi_l)_n$ corresponds to the real pixel χ_i.

Since C is a weighted selection matrix, so is C_l. Since C has full row rank, it follows from (5.23) that

$$G_y \;=\; C_l G_x C^T (C C^T)^{-1}. \tag{5.25}$$

While this expression for G_y is correct, its simple, sparse structure is obscured. However, once we note that $(CC^T)^{-1}$ is a diagonal matrix and that the weights in C_l are the same as those in C, it follows fairly easily that G_y is a lifting matrix, just as G_x is. This general result is consistent with our simple example: we define lifted-domain observations to exist for those lifted-domain nodes where corresponding original-domain observations exist, and we then assign values to these lifted measurements by simply replicating the appropriate original-domain measurement values.

The construction of R_l is facilitated by defining $\mathcal{S}(j)$ to be the set of finest-scale points, in the overlapped domain, that correspond to pixel j in the original or image domain; equivalently, $\mathcal{S}(j)$ can be defined as the number of 1s in the j-th column of G_y:

$$\mathcal{S}(j) \equiv \{i;\ G_y(i,j) = 1\}.$$

Then, consistent with this definition, $|\mathcal{S}(j)|$ is the number of times the j-th original-domain measurement is replicated. We then define R_l, the covariance of the measurement noise vector v_l, to be a diagonal matrix, where the ith diagonal entry $R_l(i,i)$ must be greater than zero and satisfy

$$\sum_{i \in \mathcal{S}(j)} \frac{1}{R_l(i,i)} = \frac{1}{R(j,j)},$$

where j is the unique index for which $G_y(i,j) = 1$. A more compact way to express this condition on R_l is to restrict R_l diagonal and positive definite, with

$$G_y^T R_l^{-1} G_y = R^{-1} \tag{5.26}$$

Any matrix R_l that satisfies these conditions provides the observation covariance amplification required in the lifted domain to offset the apparent increase in information caused by the replication of measurement. A specific choice for R_l that is consistent with (5.26) is to set

$$R_l(i,i) = |\mathcal{S}(j)|\, R(j,j), \tag{5.27}$$

where, again, j is the unique index for which $G_y(i,j) = 1$. This choice for R_l has already been seen in the context of our simple earlier example. Furthermore, this type of amplification scheme for R_l is what we will exclusively use in our numerical experiments in Section 5.6.

We have the following Proposition.

Proposition 7 *Let χ be a random field with covariance P_χ and let $y = C\chi + v$ be a set of measurements with C a weighted selection matrix and R, the covariance of v, diagonal. Suppose we then choose G_x, H_x, G_y, C_l and R_l as just described. Then the optimal linear least-squares estimate $\hat{\chi}$ of χ based on y can either be computed directly or by lifting, performing optimal estimation in the lifted domain, and then projecting. That is, if $\hat{\chi} = Ly$, and $\hat{\chi}_l = L_l y_l$, then*

$$P_\chi C^T \left(C P_\chi C^T + R\right)^{-1} = L = H_x L_l G_y = H_x P_{\chi_l} C_l^T \left(C_l P_{\chi_l} C_l^T + R_l\right)^{-1} G_y \tag{5.28}$$

where P_{χ_l} is defined in (5.8). Moreover, if $P_{\tilde{\chi}}$ denotes the estimation error covariance in estimating χ based on y, and $P_{\tilde{\chi}_l}$ denotes the estimation error covariance in estimating χ_l based on y_l, then

$$P_{\tilde{\chi}} = H_x P_{\tilde{\chi}_l} H_x^T \tag{5.29}$$

A detailed proof of this proposition is contained in Appendix D.

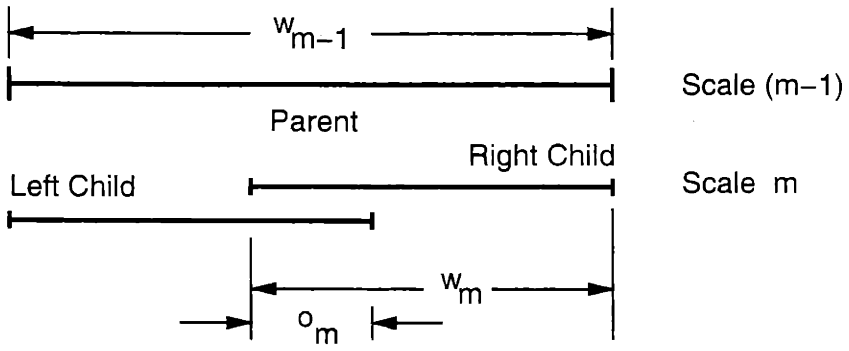


Figure 5-4: Basic overlapping-tree notation: o_m represents the degree of overlap between the regions represented by sibling multiscale nodes on scale m; w_m represents the width of the region represented by each node on scale m.

5.5 Specification of the Overlapping Framework

In this section we describe an implicit, compact, and efficient method for specifying the operators G_x and H_x. The details of this material can get quite involved, and so for simplicity, we focus on a simple case that conveys only the main ideas. A much more detailed and general description of the ideas contained in this section can be found in [23].

The case on which we focus is the representation of 1-D random processes with *dyadic* overlapping tree models. For simplicity, we impose the constraint that the structure of the overlap be spatially uniform; more specifically, we insist that for any two nodes s_1 and s_2 on the same scale of the tree, the manner in which their descendants overlap must be the same. For a dyadic overlapping tree model having this prescribed structure and also having $M+1$ scales, it turns out that the operators G_x and H_x can be specified *completely* in terms of only M parameters.

To describe this compact parameterization, we first recall that each node on the multiscale tree is associated with a connected interval of points in the original domain. We denote the width of this interval, for a node at scale m, by w_m. This convention is illustrated in Figure 5-4. The figure additionally illustrates the geometry of the overlap of the intervals associated with the two children of any given node; we denote the amount of this overlap between sibling nodes at scale m by $o_m \geq 0$.

To avoid a situation in which successive scales have the same resolution, we impose the constraint that sibling nodes do not completely overlap:

$$0 \leq o_m < w_m, \qquad m = 1, 2, \ldots, M. \tag{5.30}$$

The width parameters w_m and the overlap parameters o_m are closely related; in fact, we can see from Figure 5-4 that they are related by the following recursion:

$$w_{m-1} = 2w_m - o_m. \tag{5.31}$$

Collectively, the M overlap parameters

$$\mathcal{O} \equiv \{o_1, o_2, \ldots, o_M\},$$

provide a complete characterization of the overlap structure of the tree.

Let us turn now to the consideration of how values for M and the overlap parameters $\mathcal{O}$ are selected, to represent a 1-D sequence of length, say, K. Clearly, the length K imposes the constraint that

$$M \geq \lceil \log_2 K \rceil, \tag{5.32}$$

where the operation $\lceil x \rceil$ yields the smallest integer that is greater than or equal to x. For any fixed M satisfying (5.32), the overlap parameters $\mathcal{O}$ are implicitly constrained by two boundary conditions on the recursion (5.31). First, each node on the finest level of the tree must correspond to a single pixel:

$$w_M = 1. \tag{5.33}$$

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

$$w_0 = K. \tag{5.34}$$

The constraints (5.30)–(5.34) still leave some degrees of freedom is specifying the overlap parameters $\mathcal{O}$. In our examples in Section 5.6, we eliminate these remaining degrees of freedom by additionally constraining the so-called fractional overlap, o_m/w_m to be approximately constant as a function of scale; the fractional overlap cannot generally be made exactly constant as a function of scale, since the parameters w_m and o_m must take on integer values.

With regard to selecting a value for M, it is clear that as the value M is increased, for a fixed value of K, the amount of overlap at each scale must also increase, in order to fulfill the boundary conditions (5.33) and (5.34). Thus, if a given applications calls for a significant amount of smoothness, then we will be compelled to use a large value for M, since a greater amount of overlap leads to greater smoothness. However, in obtaining this greater smoothness, we pay a price in computational complexity, because as M increases, the complexity of carrying out simulation and and estimation also increases. Thus, there is tradeoff involved in choosing a value for M that is typically best resolved by a combination of engineering judgment and numerical experimentation.

The value of the projection matrix G_x follows uniquely, once values for M and the overlap parameters $\mathcal{O}$ have been devised. To see this fact, let us consider the k-th row of G_x, for any k. Thanks to the constraints on G_x that were established in Section 5.4.1, we know that this k-th row will have a single non-zero entry, where the value of the nonzero entry is unity. If we let s_k denote the k-th node at the finest scale of our overlapping tree, then this node will correspond to some index l_k in the 1-D process being represented, and so

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

As we now show, the index l_k can be determined directly from M and $\mathcal{O}$. Indeed, let us again consider the finest-scale node s_k. Clearly, there is a unique path from the root node 0 to the node s_k, where this path can be described as a sequence of M downward-shift

operations:

$$s_k = 0\alpha_{j_1}\alpha_{j_2}\ldots\alpha_{j_M} \qquad j_m \in \{1,2\}. \tag{5.35}$$

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

$$k = \sum_{m=1}^{M}(j_m - 1)2^{M-m}. \tag{5.36}$$

But it is easy to see from our earlier discussion of overlap geometry that the index k_l corresponding s_k must satisfy

$$l_k = \sum_{m=1}^{M}(j_m - 1)(w_m - o_m). \tag{5.37}$$

Thus, as claimed, k_l can be determined from M and $\mathcal{O}$. Since the same procedure can be repeated for every row of the matrix G_x, the entire matrix can be determined from M and $\mathcal{O}$.

The construction of H_x, while constrained by the specification of $M, \mathcal{O}$ and the fact that $H_x G_x = I$, still has degrees of freedom that must be specified. To uniquely define H_x, we recall that each component of $H_x\chi_l$ is supposed to represent the value of a pixel in the original image domain, where we insist that this value be a weighted average of the components of χ_l corresponding *only* to that single original-domain pixel. In this sense, the operator H_x performs purely an ensemble average, with *no* spatial averaging of any kind. To enforce our restriction that H_x perform no spatial averaging, we impose the condition that the distribution of non-zero elements in H_x be the same as in G_x^T; in other words, if $G_x(i,j) = 0$, then we must also have that $H_x(j,i) = 0$. One valid way to fulfill this constraint is to let H_x be equal to the Moore-Penrose pseudo-inverse of G_x; this choice additionally satisfies our constraint that $H_x G_x = I$. However, as we describe next, it is possible to devise a matrix H_x that actually does a better job of smoothing.

To describe the H_x that we actually use, let us consider two nodes on some scale m, such as the two child nodes shown in Figure 5-5(b); with these nodes fixed, let us now consider some pixel that lies within the overlapping regions of these two nodes (e.g., the pixel marked $\star$ in the figure). We need to specify the contributions of the two child nodes (and their descendants) in determining the value of pixel $\star$; for example, as indicated in the figure, the left child is given a weight of $\frac{1}{4}$ and the right child a weight of $\frac{3}{4}$. Thus the right child (and its descendants) will have a contribution three times that of the left child. In order to maintain a total contribution of unity at each pixel, we will normalize the contributions at each pixel to sum to one; these normalized values will be referred to as *relative* contributions. We propose to achieve smoothness in H_x by tapering the relative contributions of a node towards zero as one approaches an overlapped end of the interval associated with the node; one such tapering is sketched in Figure 5-5(a).

The previous paragraph outlined a procedure for determining the relative contributions of two overlapping nodes. Suppose this procedure has been applied to all nodes on all scales. To illustrate how H_x is determined in terms of these contributions, we consider a node s_k on the finest scale, and we define k and l_k as in (5.36),(5.37). The participation of node s_k on the finest scale is determined as the product of all relative contributions associated with position l_k on all ancestors of s_k. This construction is illustrated in Figure 5-6; the figure

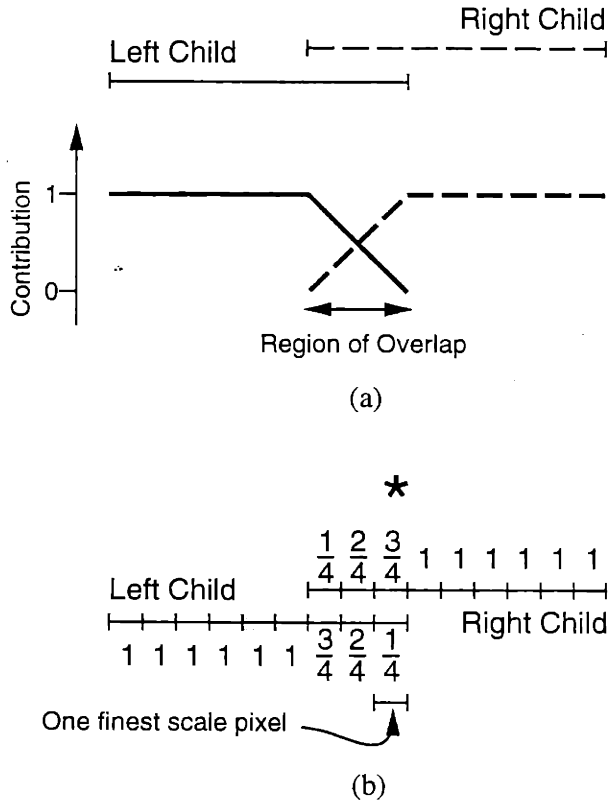
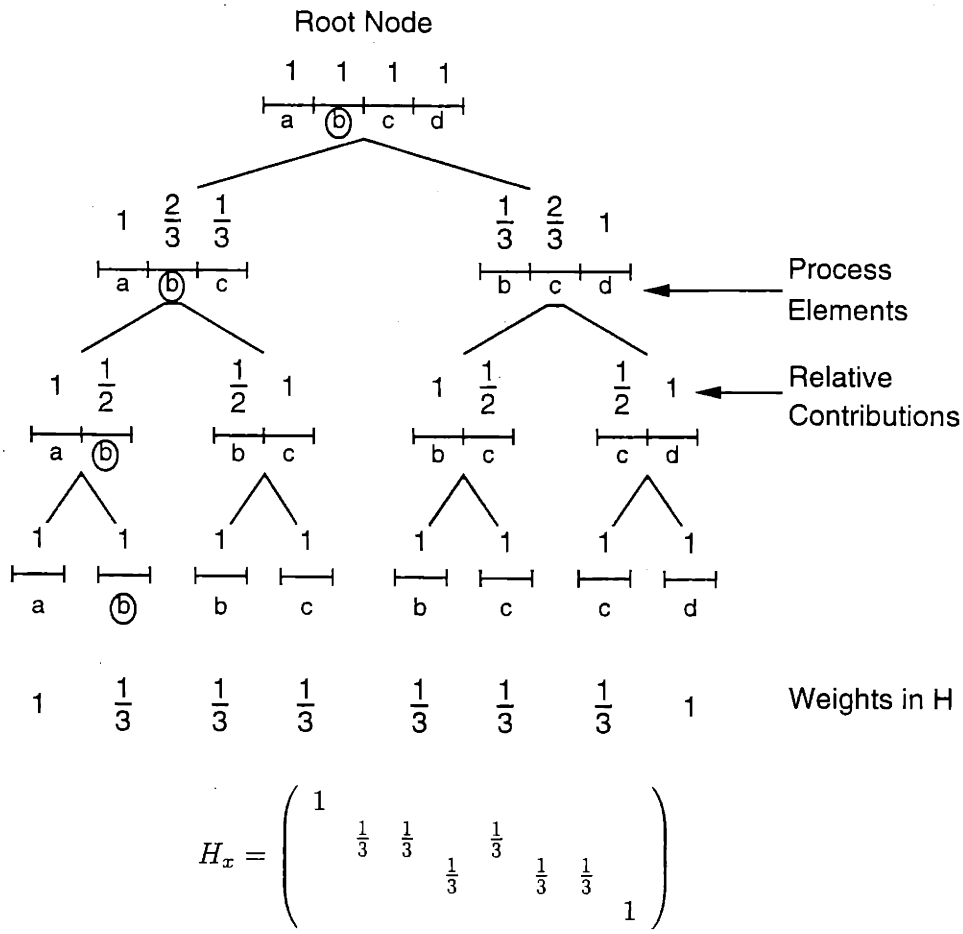


Figure 5-5: Two overlapping nodes: the set of relative contributions to each finest-scale pixel must sum to one. The contributions are tapered linearly over the region of overlap. Figure (a) shows this tapering pictorially; Figure (b) provides a specific example for two nodes which overlap by three pixels.



$$H_x = \begin{pmatrix} 1 & & & & & & & \\ & \frac{1}{3} & \frac{1}{3} & & \frac{1}{3} & & & \\ & & & \frac{1}{3} & & \frac{1}{3} & \frac{1}{3} & \\ & & & & & & & 1 \end{pmatrix}$$

Figure 5-6: An example of the construction of H_x. A four-level tree is used to represent a process having four points (a, b, c, d). The process points associated with a multiscale node are indicated below the node. The relative contributions of each node to its associated process points are indicated above each node. Products of these relative contributions determine the elements of H_x.

illustrates an overlapping tree representation of a process having four points: (a, b, c, d). Consider finest scale node $s =$ⓑ (second from the left end of the tree). The participation of s in determining the value at point b is given by the product of the relative contributions to b of all ancestors of s — i.e., the numerical values above each ⓑ in Figure 5-6. Thus the participation of s is equal to $1 \cdot \frac{1}{2} \cdot \frac{2}{3} \cdot 1 = \frac{1}{3}$; so the weight in H_x associated with s is $\frac{1}{3}$. The weights in H_x corresponding to each of the finest-scale nodes are shown in Figure 5-6.

For all but the smallest estimation problems, a dense representation of the G_x and H_x matrices is completely impractical. The observation that each row of G_x and each column of H_x contains only one non-zero entry suggests that a sparse representation based on storing only these non-zero entries might be adequately compact. However for large multidimensional problems even this sparse representation may be very large (indeed, the combined number of non-zero entries in G_x and H_x may exceed the number of values in the entire multiscale tree). As has been discussed in this section the overlapping structure parameterization $\{M, \mathcal{O}\}$, in which there are only M parameters, forms a sparse and implicit representation of G_x and H_x. We have found the construction of G_x and H_x from the overlap parameters $\mathcal{O}$ to be so rapid that we have exclusively used this latter representation in our software.

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

Table 5.1: Autoregressive weights $\{r_{k,l}\}$ of the "wood texture" WSMRF [39].

5.6 Experimental Results

The overlapping tree framework provides a powerful, new approach for carrying out both modeling and estimation. In this section we consider four applications of this framework to problems involving Markov random fields.

We focus, in particular, on a WSMRF having a fourth-order neighborhood structure and an autoregressive representation (2.8) that uses the weights given in Table 5.1 [39]. Just as in our WSMRF example in Section 3.6.2, we define the field on a toroidal lattice (of dimension 64×64), so that exact calculations, based on FFT techniques, are computationally feasible. Figure 5-7a displays a sample path of the field, drawn from the exact distribution using Gaussian deviates. Clearly, this so-called *wood* texture exhibits considerable anisotropy, having much stronger correlation in the vertical direction than in the horizontal one. This long-range vertical correlation, together with the quasi-periodic structure in the horizontal direction, are the principal qualitative features that have previously been found difficult to preserve with non-overlapping tree models, even using relatively high-order ones [45]. In contrast, we will find that our overlapping approach yields impressive results with low-order models.

All of the multiscale models used in this section are indexed on the quadtree. Since the original domain has dimension 64×64, it immediately follows that corresponding non-overlapping models will be indexed on a tree having 7 scales. On the hand, for the overlapping models, we have more freedom, as described in detail in the previous section. For all the examples in this section, we use an overlapping tree having 8 scales; the associated overlap parameters $\mathcal{O}$ have the following values:

$$\mathcal{O} = \{10, 5, 3, 2, 1, 0, 0\}.$$

We note that this choice for $\mathcal{O}$ is consistent with (5.33) and (5.34), and renders approximately constant the fractional overlap o_m/w_m. The dimension of the state vectors in the models we build will be dependent on the particular example at hand; we will make explicit this model order in each individual example. As a final note, the techniques described in Chapter 3 have been used to construct all of the multiscale models used in these examples.

5.6.1 Sample-path Generation of WSMRF

While we expect that the principal uses of our multiscale models will be for use in the design of estimation algorithms, we begin here with an example of simulating random fields. In Figure 5-7b and c, we display, respectively, sample paths associated with non-overlapping and overlapping tree models. The model orders have been chosen so that the computational effort required to generate a sample path is roughly the same for these two models; while

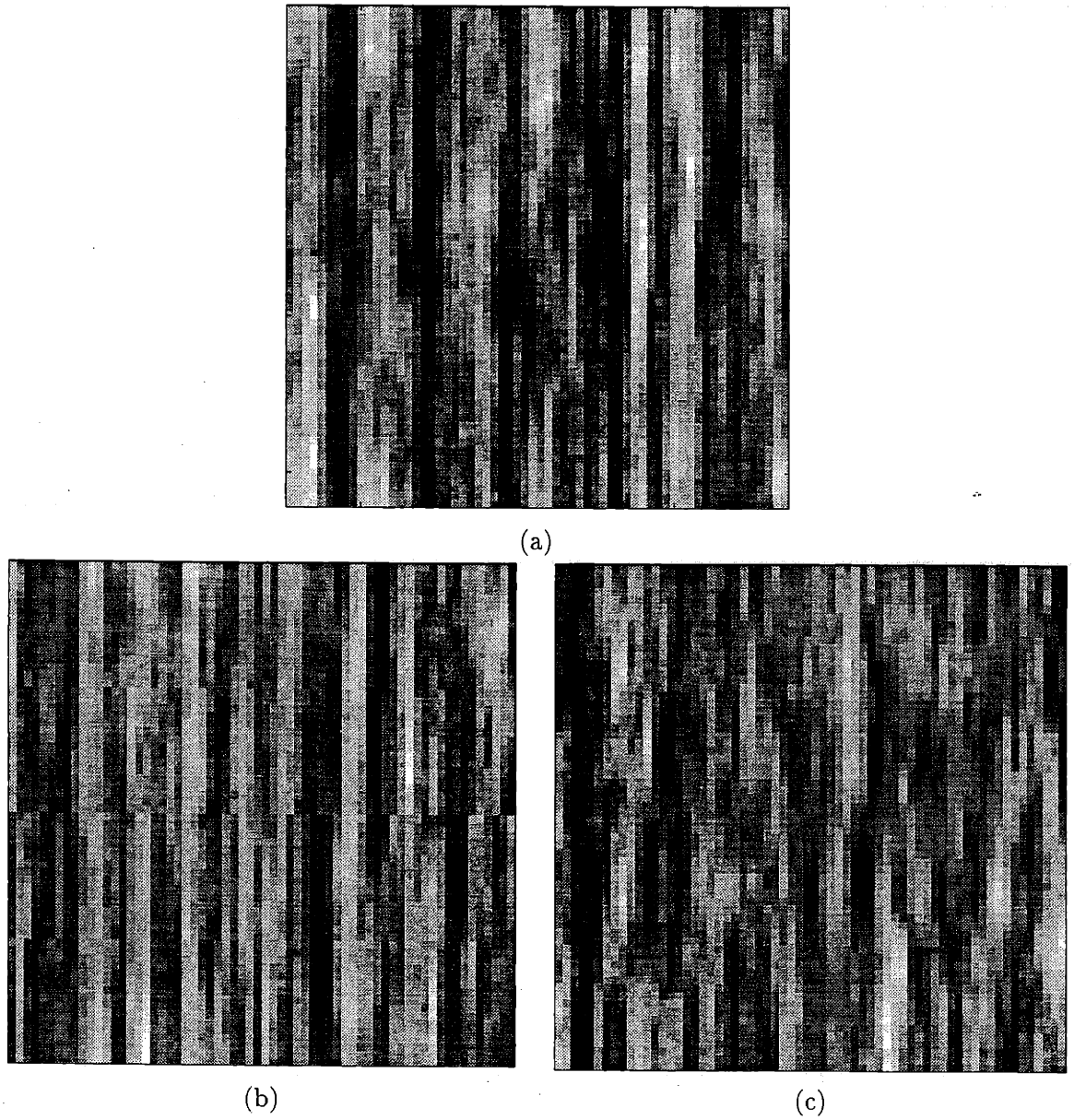


Figure 5-7: These three figures display sample paths of the wood-texture WSMRF, for a 64×64 pixel region. The sample path in (a) is drawn from the exact distribution, using FFT techniques. The sample path in (b) is based on a simulation using a non-overlapping tree model of order 64. Finally, the sample path in (c) is based on a simulation using an overlapping tree model of order 16. We note that while the sample path in (b) has noticeable blockiness, the one in part (c) has none.

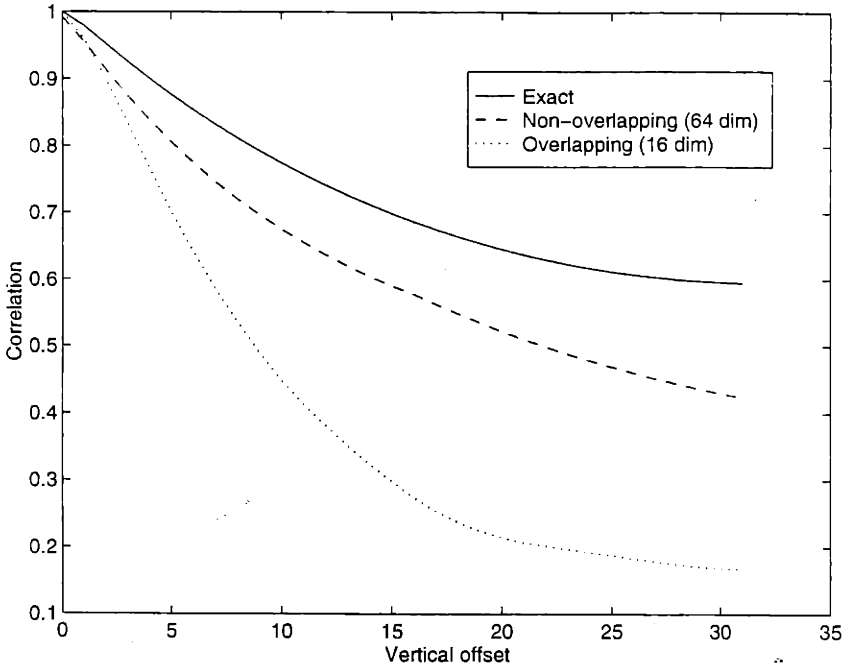


Figure 5-8: Comparison of the fidelity of a non-overlapping and an overlapping tree model for the wood texture. The solid curve displays the desired correlation function for the vertical direction. The dashed (dotted) curve displays the correlation function associated with the non-overlapping (overlapping) model; these two curves have been calculated using Monte Carlo simulation, using enough trials so that with 95% confidence, all calculated values are within 0.005 of their actual value; this uncertainty is on the order of the width of the plotted lines, and so no error bars are needed. While the overlapping model certainly yields less blocky sample paths, the non-overlapping one does a better job of preserving the ideal correlation structure in the vertical direction.

the non-overlapping model has order 64, the overlapping one has order 16. The sample path in (b) clearly suffers from visually distracting blocky artifacts, all of which are absent in the sample path in (c). In this sense, the overlapping model appears to have achieved superior performance.

In Figure 5-8, we compare the correlation functions, for the vertical direction, of the two multiscale models. Here, the non-overlapping model has achieved superior performance.

5.6.2 Estimation: Densely Sampled Field, Homogeneous Model

Our remaining three examples address various issues related to multiscale-based estimation. Our first such example is for a case in which we have dense, regularly sampled measurements of uniform quality. These conditions allow us to compare our multiscale-based estimates with statistically optimal, FFT-based estimates, where the latter can be efficiently calculated in this special case. To carry out the experiment, we have corrupted the original sample path displayed in Figure 5-7a with white Gaussian noise, to yield an observed image having 0dB SNR (i.e., the variance of the measurement noise is equal to the variance of the signal). In Figure 5-9, we then compare three different estimates of the original field, based on the given noisy image. In (a), we display the estimate produced by optimal FFT-based techniques, while in (b) and (c), we display the estimates associated with non-overlapping and overlapping tree models, respectively. Again, we have chosen the multiscale model orders so that the computational effort required to calculate (b) and (c) is roughly the same;

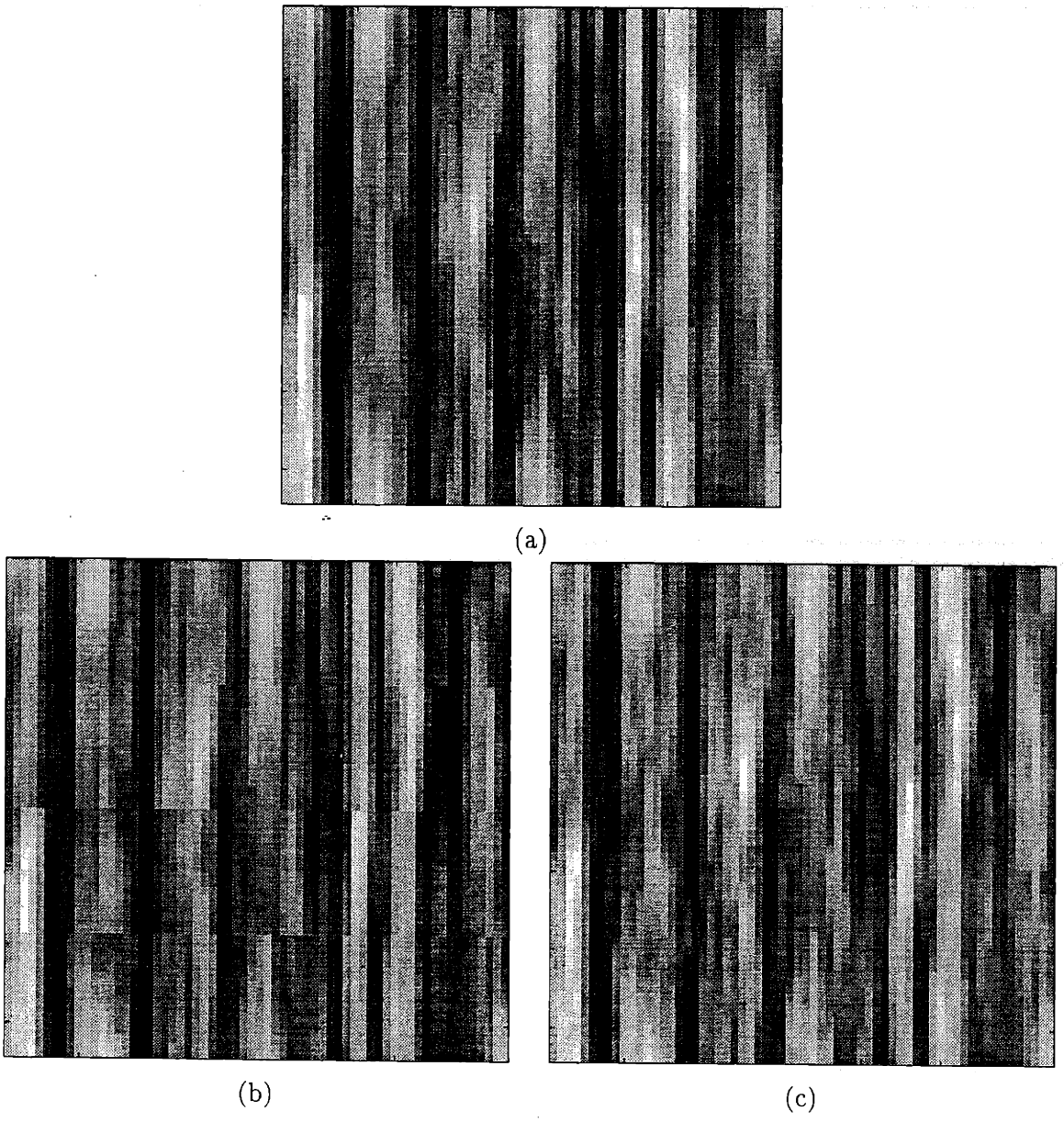


Figure 5-9: These three figures display linear least-squares estimates of the sample path in Figure 5-7a, based on dense, noisy measurements of the signal in Figure 5-7a with 0dB SNR. (a) The statistically optimal estimate, which is calculated using FFT techniques. (b) A multiscale-based estimate, using a non-overlapping tree process in which the state dimension is constrained to be no greater than 40. Note the blocky artifacts at the quadrantal boundaries. (c) A multiscale-based estimate, using an overlapping tree process, in which the state dimension is constrained to be no greater than 16. Note that all blocky artifacts have been eliminated. The computational burden is the same for computing the estimates in (b) and (c).

the non-overlapping model has order 40, while the overlapping one has order 16.

One way to compare these two multiscale-based estimates is in terms of percentage loss in error-variance reduction; this metric was defined in (3.45). By this criterion, the non-overlapping estimate in (b) has a loss of 1.4%, while the overlapping estimate in (c) has a loss of 2%. Hence, with regard to this criterion, the non-overlapping estimate has done slightly better for this one trial. On the other hand, a *visual* inspection of these estimates reveals that while the one in (b) has distracting blocky artifacts, the one in (c) does not. Thus, if the elimination of such artifacts is an important concern, using an overlapped model is decidedly superior.[6] Furthermore, if a MSE closer to the optimum is also desired, the use of a model of slightly higher dimension can achieve that as well.

Although the FFT technique is both efficient and optimal in terms of MSE, it suffers from a limited applicability to special circumstances. In particular, each of the following cases precludes the use of the FFT, but may be solved using our multiscale method: (i) irregularly sampled measurements, (ii) spatially varying measurement noise, and (iii) spatially varying prior model. Indeed, we next examine estimation problem for which the FFT-based techniques are inapplicable.

5.6.3 Densely Sampled Field, Heterogeneous Model

We now consider an estimation problem for which FFT techniques are inapplicable: the computation of estimates for a random field having a non-stationary prior model. Figure 5-10a displays a sample path of the non-stationary model. The 64×64 pixels of the field were divided into groups g_1 and g_2: g_1 contains the pixels in the upper left and lower right of the image, and g_2 contains the pixels in the diagonal band running through the center of the image. The prior model for the pixels in g_1 is the "wood" MRF model of Table 5.1; the prior model for the pixels in g_2 uses the the same coefficients in Table 5.1, but with the whole table rotated by 90 degrees. The cross correlation between groups g_1 and g_2 is zero.

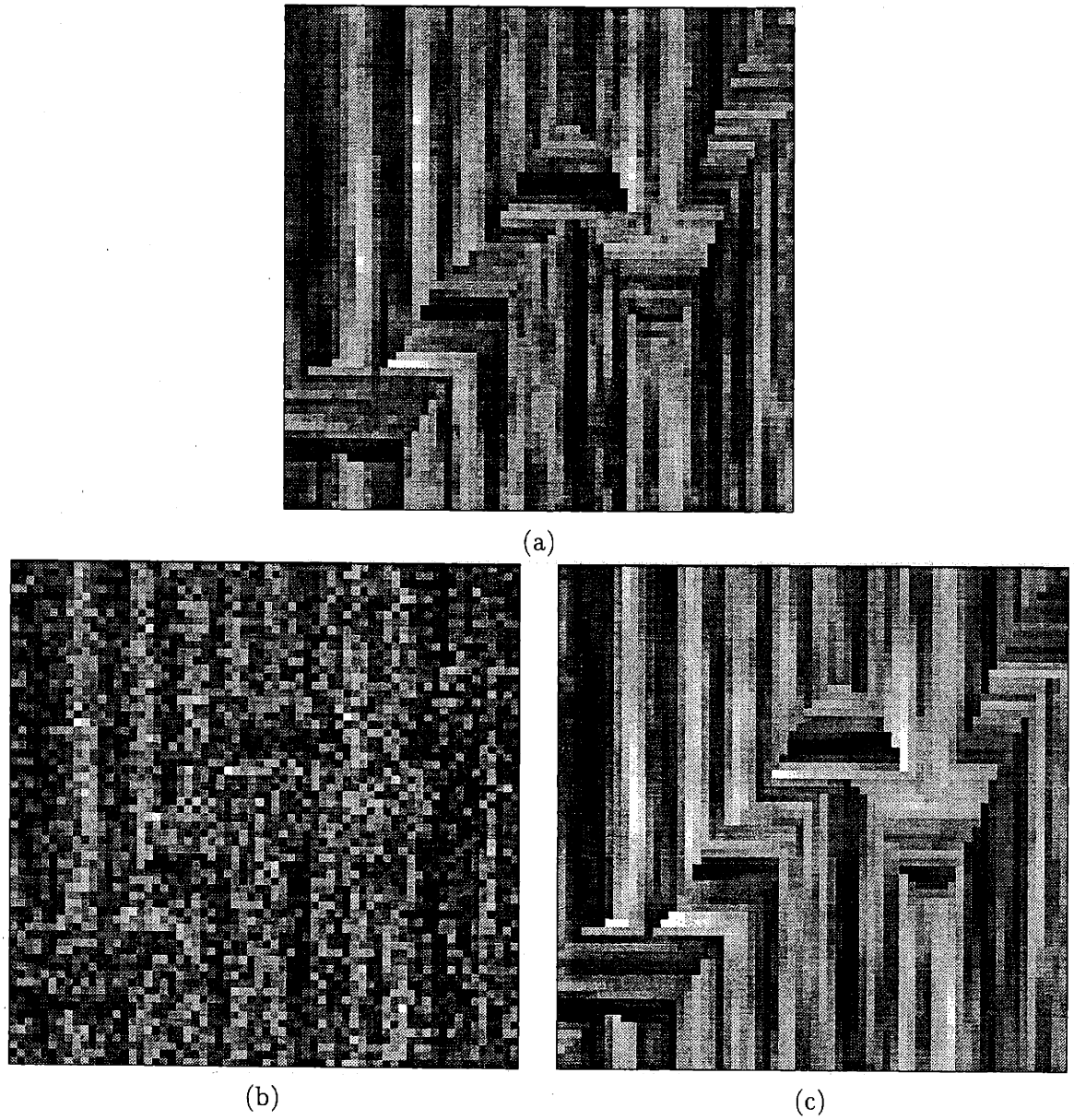
The choice of such a non-stationary prior model, as opposed to the simple prior model of the previous example, just implies a change in the prior statistics on the finest scale of the multiscale tree. Otherwise there is no essential difference, and the multiscale model development and estimation procedure proceed unaffected.

Figure 5-10b displays a noisy version of the original sample path, corrupted by white Gaussian noise to 0dB; Figure 5-10c displays the corresponding multiscale reconstruction based on an overlapping multiscale model of order 32. As mentioned in the previous example, the smoothing operation H_x of the overlapping framework has not at all blurred the edge between the two prior models — the edge stands out distinctly. Furthermore, no blocky artifacts are apparent anywhere in the reconstruction.

5.6.4 Locally Sampled Field, Homogeneous Model

Let us consider another estimation problem in which FFT techniques are inapplicable: the computation of a set of estimates given a stationary prior model, but with measurements available at only a small, non-rectangular subset of the pixels. Figure 5-11a shows a subset of the pixels of the "wood" texture from Figure 5-7a; this elliptical set of pixels represents

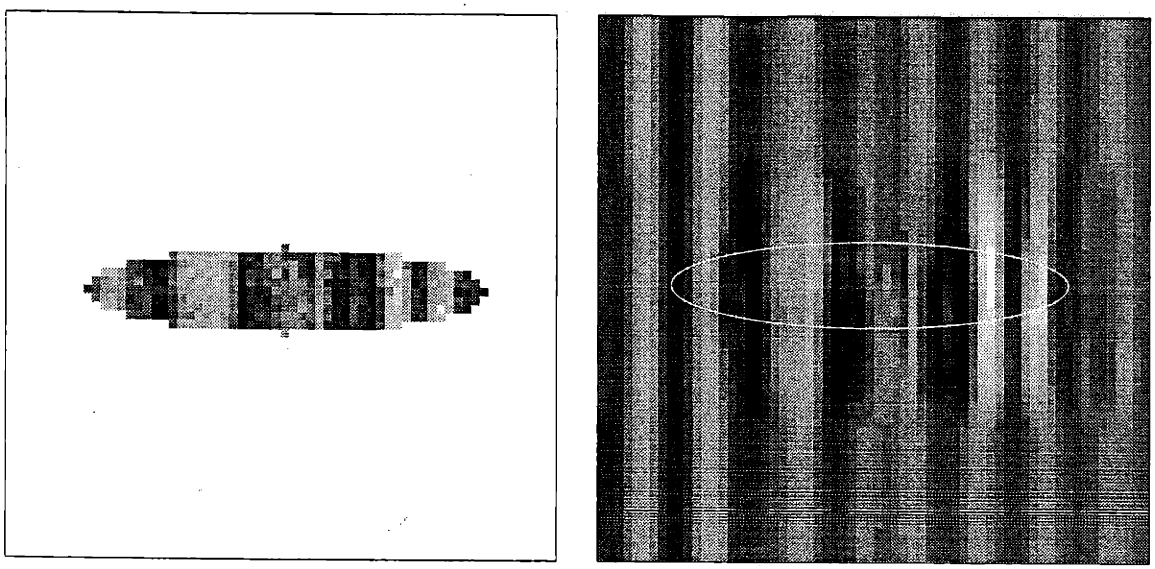
[6] This points to the fact that MSE is not always the criterion by which we judge reconstructions. In fact, in the eyes of both this thesis author and his collaborators, the reconstruction in Figure 5-9c is superior not only to the one in Figure 5-9b, but also to the optimal FFT-based reconstruction Figure 5-9a, which seems excessively smooth compared to the original process in Figure 5-7a.



(a)

(b) (c)

Figure 5-10: These three figures display results of linear least-squares estimation with a heterogeneous texture. (a) A sample path of an inhomogeneous MRF, in which each pixel belongs to either a horizontally or vertically correlated texture. (b) Observation of the sample path in (a), with corruption by 0dB white, Gaussian noise. (c) Estimate of the sample path in (a), based on the observation in (b), using overlapping tree model of order 32.



(a) (b)

Figure 5-11: These two figures display results of linear least-squares estimation based on observation of only part of the field. (a) Noiseless observations of small subset of the sample path in Figure 5-7a. (b) Estimate of the sample path in Figure 5-7a, based on the observation in (a), using an overlapping tree model of order 16. We note that the structure of the estimate has a very smooth evolution from being grainy and detailed to being smooth and less detailed, as the distance increases from the center of the ellipse of observations.

those pixels to be used as measurements. No noise was added to the measurements, however since a measurement error variance of zero is not permitted in the particular implementation of the multiscale estimator used here,[7] a measurement noise variance of 10^{-4} was specified.

Being given measurements at a subset of the image pixels, as opposed to a dense set of measurements as in the previous two examples, just implies a trivial change in the measurement projection operator G_y and, consequently, in the multiscale measurement matrices on the finest scale of the tree. Otherwise there is no essential difference, and the multiscale model development and estimation procedure proceed unaffected. It is rather significant to note, however, that while the multiscale framework is readily adapted to the loss of measurements, a change from dense to irregular sampling immediately makes FFT-based approaches inapplicable.

Figure 5-11b displays the overlapping tree reconstruction, based on the limited set of measurements given in part (a) of the figure. The multiscale estimator does capture the coarse features of the original texture of Figure 5-7a outside of the measured region. Even certain aspects of the vertical bands to the left and right of the measured region are properly captured. Also, once again, despite the fact that we are using a multiscale estimator, the estimated texture evolves smoothly, without blocky artifacts, as we move away from the measured pixels.

For our final estimation example, we again consider estimation using a stationary prior model, but now with noiseless observations that are spatially distributed according to a sample path of 2-D Poisson process. In Figure 5-12a, we display the original sample path, while in part (b), we display the locations of the observations (denoted by the blackened pix-

[7] Although such an estimator *could* be implemented.

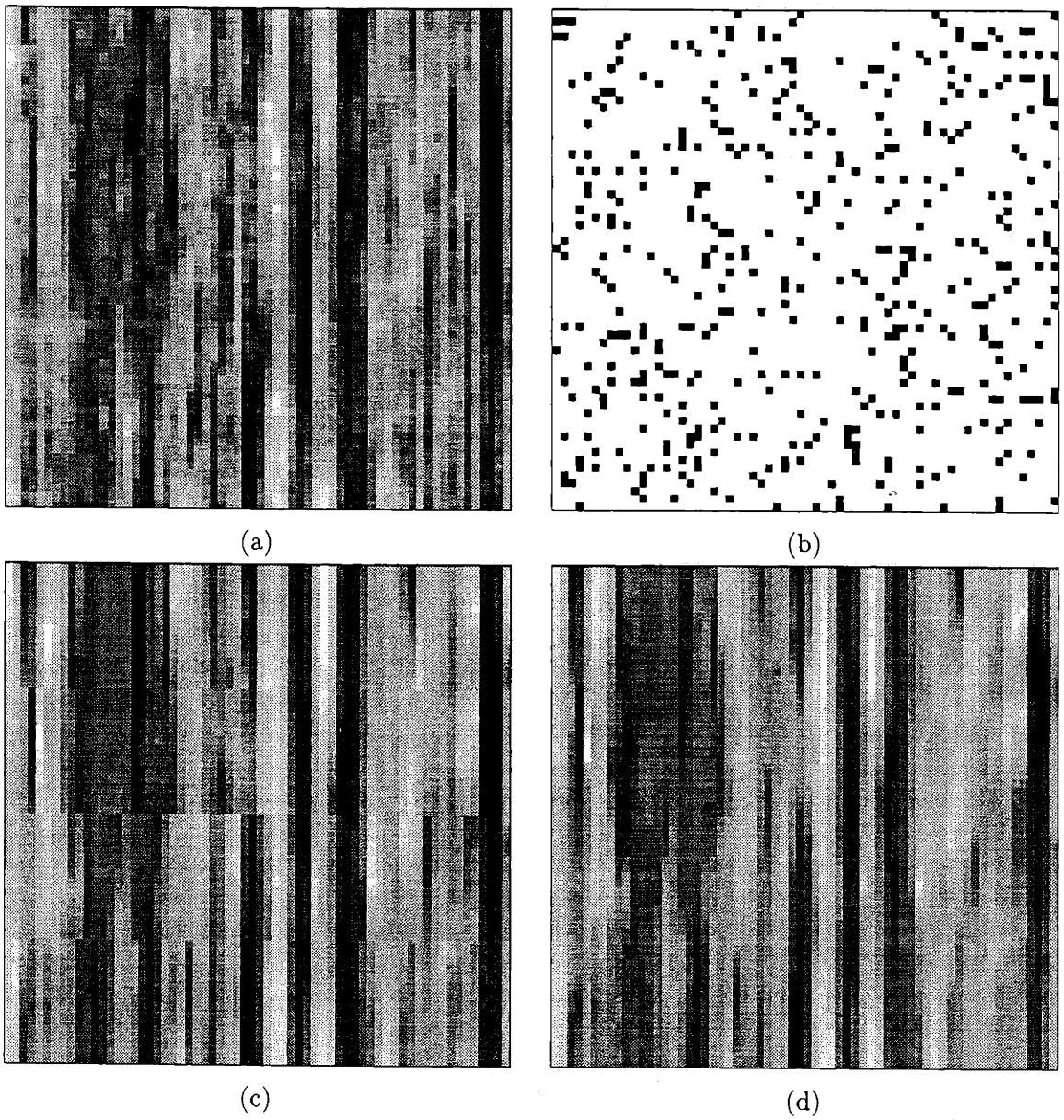


Figure 5-12: These four figures display another example of linear least-squares estimation based on observation of only part of the field. (a) Sample path of original field. (b) Locations of observed pixels; these observations represent only 10% of all the pixels. (c) Estimate of the signal in (a), based on the observations in (b), using a non-overlapping model of order 40. (d) Estimate of the signal in (a), based on the observations in (b), using an overlapping model of order 16.

els.) In Figures 5-12c and d, we display, respectively, estimates based on a non-overlapping and an overlapping model. The order of the non-overlapping (overlapping) model is 40 (16). In this case, the estimate based on the overlapping model is both more visually pleasing *and* has a lower mean-square error. In particular, the MSE of the estimate in (c) is 0.1563, while the MSE of the estimate in (d) is 0.1440; the pixel variance in the original image (in part (a)) was unity.

5.7 Conclusion

We have presented a new approach to modeling and estimation, using a recently introduced class of multiscale stochastic processes. Our work has been motivated by the observation that estimates based on the types of multiscale models previously proposed have tended to exhibit a visually distracting blockiness. To eliminate this blockiness, we have discarded the standard assumption that distinct nodes on a given level of the multiscale process must correspond to disjoint portions of the image domain. Instead, we allow distinct tree nodes to correspond to overlapping portions of the image domain, thereby eliminating the hard boundaries between pixels. This is done in a way that eliminates blocky artifacts *without* spatial averaging, so that if a field does indeed have sharp discontinuities, these can be captured without blurring in our framework.

By coupling this overlapping framework with a multiscale stochastic realization tools of Chapter 3, we have developed a powerful estimation and modeling tool which allows one to manage the tradeoff among estimate smoothness, statistical fidelity, and computational effort. Furthermore we have characterized the optimality properties of our estimation procedure. As we have discussed, these properties imply the important result that any actual suboptimality is traceable completely to approximations made in building a low-dimensional model for the specified statistics. We thus have explicit control of the complexity-accuracy tradeoff.

In the examples of Section 5.6, we applied the overlapping multiscale framework to problems of modeling and estimation involving MRFs. The flexibility of the multiscale framework allows us to confront problems for which FFT techniques are not applicable; in particular, we considered problems involving nonstationary statistics, where we found that edge preservation was possible, and also problems involving irregularly sampled data. Actually, the flexibility of our framework is greater than that implied by examples considered here; in particular, the modeling and estimation of processes in higher dimensions is also possible.

Chapter 6

Conclusions and Suggestions for Future Research

This chapter summarizes the contributions of this thesis and provides some perspective on and suggestions for future research.

6.1 Thesis Contributions

The focus of this thesis has been on extending, refining and applying a recently introduced framework for multiscale stochastic modeling. In Chapters 1 and 2, we laid the foundation for this work by providing a broad view of the challenges of statistical inference with 2-D random fields and by reviewing previous results that are needed to understand our specific research contributions. Subsequently, in Chapters 3, 4 and 5 we pursued the core of our development, where our specific, new contributions were detailed. Here, we summarize and review these contributions.

6.1.1 A Theory for Multiscale Stochastic Realization

In Chapter 3, we developed elements of a theory for multiscale stochastic realization. We focused in particular on the problem of building multiscale models to realize, either exactly or approximately, prespecified finest-scale statistics. In this context, we formalized the reduced-order modeling problem, we developed model-building algorithms for addressing this problem, and we demonstrated the viability of our approach in an extensive set of numerical experiments. The specific contributions of this work can be summarized as follows.

First, we have successfully brought to bear a popular reduced-order modeling tool from the time-series context to the multiscale context. This tool is canonical correlation analysis, which was originally developed in multivariate statistics as a method for displaying unambiguously the correlation structure between two random vectors. Akaike adapted this tool to the dynamical context of state-space realization of time series, where he used it to devise particular bases for state vectors, in which the components of the state are arranged in descending order of importance to the past/future interface. This arrangement allows for a straightforward, rational decision about which components of the state to discard in a reduced-order realization. We have made a further extension of canonical correlation analysis to the multiscale context, allowing us to build rationally and systematically

reduced-order multiscale models. Our use of canonical correlations represents a non-trivial extension of the time-series approach, primarily because of the complications that arise when the process state must act as an interface among three or more subsets of the process.

Second, we have demonstrated the utility of our modeling methods by considering a number of numerical examples, involving both 1-D random processes and 2-D random fields. These models allowed us to confront some challenging 2-D estimation problems (involving, for example, isotropic random fields) that are impractical to address with more traditional, FFT-based estimation methods.

Third, we have highlighted some interesting differences between time-series stochastic processes and multiscale stochastic processes. While one difference is the two-way versus multi-way nature of the information interface provided by the process state, there are others. Most notable is the difference in the richness of the class of internal realizations. In particular, in the time-series context, the class of internal realizations is sufficiently rich to contain minimal, exact realizations of specified statistics. On the other hand, multiscale internal realizations do not enjoy the same richness; there sometimes exists a so-called external realization of given statistics that has lower dimension than any internal realization of the same statistics. We illustrated this latter fact by considering in detail a specific example.

Finally, we have brought into sharper focus the issues that must be confronted in the multiscale modeling problem. We have seen, for example, that one attractive features of our primary modeling approach is its decomposition of the problem into a collection of independent sub-problems that can each be solved myopically. However, we further noted that this computational attractiveness comes at a prices; in particular, the approach's myopia leads to a sacrifice of tight control over the interscale propagation of state information. We examined a specific example to highlight this fact, and we then developed an alternative model-building approach that handles the propagation of information more explicitly. In its present form, the computational cost of this alternative is prohibitive for problems of practical size. Nevertheless, this alternative provides a nice contrast to our primary approach, and together the two illustrate the tradeoffs involved in the design of multiscale model-building algorithms.

6.1.2 A Multiresolution Approach to Discriminating Targets from Clutter in SAR Imagery

In Chapter 4, we identified multiscale models for high resolution, millimeter-wave SAR imagery. Subsequently, we used these models to facilitate likelihood calculations in an important discrimination problem in ATR; more specifically, we used our identified models to define a multiresolution discriminant as the likelihood ratio for distinguishing between man-made objects and natural clutter, given a multiresolution sequence of images of a region of interest. We then incorporated this likelihood ratio into an existing, established discriminator that was developed at Lincoln Laboratory as part of a complete system for automatic target recognition (ATR). Finally, we tested this modified ATR system on an extensive dataset of actual SAR imagery, and compared the performance to that of the established, traditional Lincoln ATR system. The specific contributions of this work can be summarized as follows.

First, although speckle has traditionally been treated as an obstacle to good ATR performance, we have joined [61] in taking and persuasively supporting the opposite view. We argue that in fact the speckle signature is dependent on scatterer type and that this depen-

dency can be exploited to improve discrimination performance. The quality of our results supports this claim.

We found that our approach to discrimination leads to a substantial and statistically significant improvement in receiver operating characteristics, compared to an optimized version of the standard discriminator that is traditionally used in the Lincoln Laboratory ATR system. For instance, at a probability of detection of 0.95, our new discriminator reduces the number of natural-clutter false alarms by almost a factor of six.

6.1.3 An Overlapping-Tree Approach to Modeling and Estimation

In Chapter 5, we extended the multiscale framework by relaxing the standard assumption that distinct nodes on a given level of the multiscale process must correspond to disjoint portions of the image domain; instead, we allowed a correspondence to overlapping portions of the image domain. Using the stochastic realization techniques of Chapter 3, we then built so-called overlapping-tree models, which we subsequently used for both sample-path generation and least-squares estimation of random fields. The specific contributions of this work can be summarized as follows.

First, our approach provides a nice way to overcome the visually distracting blocky artifacts that are typical of sample paths and estimates produced by standard multiscale models. Although in some applications, these artifacts are unimportant and are completely lacking in statistical significance [46], in other cases [24], multiscale-based estimates are subsequently used in a manner that requires the calculation of surface gradients; in these latter cases, there is an essential need for having smooth estimates, so that the gradients can be calculated meaningfully.

Although estimate blockiness can be eliminated by simpler means than our overlapping-tree approach, (i.e., by post-processing with a low-pass filter), such simplicity comes at a significant price. In particular, low-pass filtering can render less clear the proper interpretation of error covariance information provided by the estimation algorithm, and the spatial blurring produced by a low-pass filter post-processor limits the resolution of fine-scale details in the post-processed estimate. In contrast, our overlapping-tree approach retains one of the most important advantages of the multiscale estimation framework, namely the efficient computation of estimation error covariances. Moreover, our numerical experiments demonstrated that we can proceed with low-dimensional multiscale models that are quite faithful to prespecified random field covariance structures to be realized. Thus, our estimates are not only smooth, but are nearly optimal in terms of mean-square error and can be calculated (together with the error covariance information) efficiently.

6.2 Suggestions for Future Work

6.2.1 A Theory for Multiscale Stochastic Realization

Relation between local and global measures of model fidelity

A large fraction of our effort in Chapter 3 was focused on determining the information content of the state vectors $x(s)$, by solving for W_s matrices to address either (3.20) or (3.21). While our numerical experiments in Section 3.6 suggest that both (3.20) and (3.21) provide useful guidelines for building reduced-order models, these criteria are only indirectly related to the overall quality of model fit. In fact, it is natural to interpret the parameter γ_s in (3.20) and the parameter λ_s in (3.21) as *local* measures of model fidelity. An interesting

and important question is how to relate analytically γ_s and λ_s to more global fidelity measures, such as loss in error-variance reduction (see (3.45)) or the *Bhattacharrya distance* (discussed later in this section).

Although Monte-Carlo simulation can certainly be used to relate experimentally our local fidelity metrics to more global ones, there are compelling reasons to characterize analytically these relationships. One reason is that analytical relations can aid in choosing uniform values for the local parameters $\{\gamma_s\}_s$ or $\{\lambda_s\}_s$. To clarify our meaning of *uniform* here, we need only consider an extreme case in which we build a multiscale model in which the root node state is constrained to have dimension no greater than $10,000$ (i.e., $\lambda_0 = 10,000$, while all the coarser-scale states are constrained to have dimension no greater than 1 (i.e., $\lambda_s = 1$, $s \neq 0$). Colloquially, a chain is no stronger than its weakest link, and so it is here with multiscale models: our hypothetical values for $\{\lambda_s\}_s$, will likely lead to very poor overall fidelity, with the high fidelity information contained in the root node having no way to propagate to finer scales. The point here is that the local fidelity should be uniform throughout a model. Moreover, to find uniform values, it would be helpful to have analytical relations between local fidelity and global fidelity.

Another reason to seek analytical relations is that they are invaluable in addressing an even deeper question regarding model fidelity: *as image size grows, how must state dimension grow to keep overall model fidelity at a constant level?* We certainly expect that the needed model order *will* grow with image size, but at what rate? At present, the only case in which we can readily characterize this rate is for exact realizations of WSMRFs, where the relation between image dimension (i.e., length or width) and model order is, by inspection, asymptotically linear. For reduced-order models, our numerical experiments suggest that model order grows more slowly than image dimension, but this observation is only anecdotal, and has not been carefully quantified. Much work remains to be done to resolve this issue satisfactorily.

Bhattacharrya distance: a global metric

In our numerical experiments, we used loss in error-variance reduction as a global measure of model fidelity. An alternative, that is less closely tied to any one application, is the *Bhattacharrya distance* [38], defined by

$$d(p_x(X), p_z(Z)) \equiv -\ln \int \sqrt{p_x(t)\, p_z(t)}\, dt,$$

for any two PDFs $p_x(X)$ and $p_z(Z)$. This distance measure, which is commonly used in information theory, is closely related to the probability of error in binary hypothesis testing problems. To make the connection precise, suppose that we have a Bayesian binary hypothesis testing problem in which the two hypotheses, H_0 and H_1, are equally likely. When H_0 is true, we observe a realization of the random vector Z and when H_1 is true, we observe a realization of the random vector X; the hypothesis testing problem is to design a decision rule that uses the observation to decide which hypothesis is true. The theory associated with the Bhattacharrya distance tells us that there exists a decision rule having a probability of error that satisfies the bounds [38]

$$\frac{1}{2}\left(1 - \sqrt{1 - \exp(2d(p_x(X), p_z(Z)))}\right) \leq \begin{matrix}\text{Probability} \\ \text{of error}\end{matrix} \leq \frac{1}{2}\exp[-d(p_x(X), p_z(Z))] \quad (6.1)$$

In many situations, the upper bound in (6.1) is quite tight [38], and in these situations, the Bhattacharrya distance behaves in a manner consistent with how we intuitively expect a distance measure should behave.[1] For example, if the distance is large, then in the context of the binary hypothesis testing problem, we expect that a reasonable decision rule should have a low probability of error, which in turn, is consistent with the error bound in (6.1). For more information about the Bhattacharrya distance, we refer the reader to [38].

To illustrate the challenges of employing this metric in our modeling context, we consider a simplified. special version of the multiscale modeling problem. Proceeding, let us suppose that we have a zero-mean, Gaussian random vector Z of $2N$ components, for which

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}, \quad E\left[\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}\begin{pmatrix} Z_1^T & Z_2^T \end{pmatrix}\right] = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{pmatrix},$$

where Z_1 and Z_2 are each N-dimensional, and both Σ_{11} and Σ_{22} have full rank. We wish to realize Z as the finest scale of a two-level multiscale process, defined on a dyadic tree. We note that there are only three nodes in this two-level dyadic-tree process, and so for this simple, special case we specialize our notation. In particular, let x_0 denote the state vector at the root node, and let its distribution be given as

$$x_0 \sim \mathcal{N}(\mathbf{0}, P_0).$$

Let x_1 and x_2 be the two state vectors at the finest scale, and let them be related to the state at the root node by the relations

$$x_i = A_i x_0 + B_i w_i \qquad (i = 1, 2),$$

where the driving terms w_1 and w_2 are independent and identically distributed Gaussian random vectors,

$$w_i \sim \mathcal{N}(\mathbf{0}, I) \qquad (i = 1, 2).$$

In terms of these notational conventions, we can precisely state our realization goals. Subject to certain conditions, we want to specify the values of the model parameters P_0, A_1, A_2, B_1, and B_2. We *insist* that these parameters be chosen such that x_1 has the same marginal statistics as z_1 and that x_2 has the same marginal statistics as z_2:

$$\begin{aligned} E(x_i x_i^T) &= A_i P_0 A_i^T + B_i B_i^T \\ &= \Sigma_{ii} \qquad (i = 1, 2). \end{aligned} \tag{6.2}$$

As we will see, these two conditions can be trivially fulfilled, and their fulfillment does not form the interesting part of the analysis. The interesting issue is the tradeoff between the dimension of the root node state x_0 and the fidelity of the preservation of the desired cross-correlation between x_1 and x_2. The primary objective of the following analysis is to explore this tradeoff.

We know, from Proposition 1, that to achieve equality between Σ_{12} and $E(x_1 x_2^T)$, the dimension of x_0 must be at least as large as the rank of Σ_{12}. In fact, by applying Corollary 1,

[1] Although, we remark that the Bhattacharrya distance *does not* obey the triangle inequality.

we readily see that one possibility is

$$P_0 = D, \tag{6.3}$$

$$A_i = T_i^+ \begin{pmatrix} I_{m_{12}} \\ \mathbf{0} \end{pmatrix} \qquad (i = 1, 2), \tag{6.4}$$

$$B_i = T_i^+ \left[I - \begin{pmatrix} D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \right]^{1/2} \qquad (i = 1, 2). \tag{6.5}$$

In these relations, we are using the notational conventions established in Section 2.4.1, with m_{12} equal to the rank of Σ_{12}. In light of (6.3)-(6.5), we are interested in precisely characterizing the Bhattacharrya distance between $p_x(X)$ and $p_z(Z)$ when the dimension of x_0 is pushed below m_{12}.

We consider a particular, structured family of approximate multiscale representations. We index these models by the dimension m_0 of the root node state; there are m_{12} models in the family, ranging from exact and complex (wherein $m_0 = m_{12}$) to very loosely approximate and simple (wherein $m_0 = 1$). Thus, we drive $m_{12} - m_0$ of the components of x_0 to zero variance. In this way, we obtain the following family of models:

$$P_0 = \hat{D}_{m_0}, \tag{6.6}$$

$$A_i = T_i^+ \begin{pmatrix} I_{m_0} \\ \mathbf{0} \end{pmatrix} \qquad (i = 1, 2), \tag{6.7}$$

$$B_i = T_i^+ \left[I - \begin{pmatrix} \hat{D}_{m_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \right]^{1/2} \qquad (i = 1, 2), \tag{6.8}$$

where the diagonal matrix $\hat{D}_k$ is a modified version of D, in which the $m_{12} - k$ smallest diagonal entries are nulled out to zero. One can readily verify that every member of this family of models preserves the marginal statistics that we want to preserve, as dictated by (6.2). Thus, these approximate representations have only degraded the fidelity of the desired cross-correlation statistics.

Under mild assumptions, the Bhattacharrya distance can be analytically evaluated for our problem [26]. To show this, we must first establish some notation. Let Σ denote desired covariance of the finest scale process,

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{pmatrix}.$$

Also, let $\hat{\Sigma}_{m_0}$ be the covariance of the finest scale process that results when we use a model from our family of approximate representations,

$$\hat{\Sigma}_{m_0} = \begin{pmatrix} \Sigma_{11} & \hat{\Sigma}_{12} \\ \hat{\Sigma}_{12}^T & \Sigma_{22} \end{pmatrix}.$$

In this latter expression, the subscript m_0 is a reminder of the dimension of the root node in the multiscale model approximate representation; also, the cross-covariance is given by

$$\hat{\Sigma}_{12} = A_1 P_0 A_2^T.$$

Finally, for our problem, we denote the Bhattacharrya distance by $d(\Sigma, \hat{\Sigma}_{m_0})$.

If both Σ and $\hat{\Sigma}_{m_0}$ are strictly positive definite, then we can analytically evaluate $d(\Sigma, \hat{\Sigma}_{m_0})$ as

$$d(\Sigma, \hat{\Sigma}_{m_0}) = \frac{1}{2} \ln \left[\frac{\left| \frac{1}{2}\Sigma + \frac{1}{2}\hat{\Sigma}_{m_0} \right|}{|\Sigma|^{1/2} \left| \hat{\Sigma}_{m_0} \right|^{1/2}} \right]. \tag{6.9}$$

Letting

$$T = \begin{pmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{pmatrix}, \tag{6.10}$$

and noting that if Σ and $\hat{\Sigma}_{m_0}$ are invertible, then so is T, we can simplify (6.9) as follows:

$$\begin{aligned} d(\Sigma, \hat{\Sigma}_{m_0}) &= \frac{1}{2} \ln \left[\frac{|T|\,|\frac{1}{2}\Sigma + \frac{1}{2}\hat{\Sigma}_{m_0}|\,|T^T|}{|T|^{1/2}\,|\Sigma|^{1/2}\,|T^T|^{1/2}\,|T|^{1/2}\,|\hat{\Sigma}_{m_0}|^{1/2}\,|T^T|^{1/2}} \right] \\ &= \frac{1}{2} \ln \left[\frac{|\frac{1}{2}T\Sigma T^T + \frac{1}{2}T\hat{\Sigma}_{m_0}T^T|}{|T\Sigma T^T|^{1/2}\,|T\hat{\Sigma}_{m_0}T^T|^{1/2}} \right] \\ &= \frac{1}{2} \ln \left[\frac{|I - \frac{1}{4}(D + \hat{D}_{m_0})^2|}{|I - D^2|^{1/2}\,|I - \hat{D}_{m_0}^2|^{1/2}} \right] \\ &= \frac{1}{2} \sum_{k=m_0+1}^{m_{12}} \ln \left(\frac{1 - \frac{1}{4}d_k^2}{\sqrt{1 - d_k^2}} \right) \\ &= \frac{1}{2} \sum_{k=m_0+1}^{m_{12}} \ln \left(\frac{3 + \sin^2 \theta_k}{4 \sin \theta_k} \right), \end{aligned}$$

where θ_k is the principal angle associated with d_k (defined by $\theta_k = \cos^{-1} d_k$), and $d_1, d_2, ..., d_{m_{12}}$ are the diagonal elements of the matrix D.

This result has a simple, intuitively satisfying interpretation. In particular, the Bhattacharrya distance $d(\Sigma, \hat{\Sigma}_{m_0})$ increases monotonically as the dimension of x_0 decreases below m_{12}. Moreover, the distance increases in such a way that we can easily pinpoint the distance contribution from each incremental drop in the dimension of x_0. These distance contributions are a monotonic function of the change in the affected principal angle between the vector spaces $\mathcal{X}_1$ and $\mathcal{X}_2$; the contribution monotonically increases from zero (if the angle realignment is zero radians) to infinity (as the angle realignment approaches $\pi/2$ radians).

The challenge remains to generalize an analysis such as this one to a multiscale process having more than two levels. This sort of analysis appears to be quite challenging, but is certainly worthwhile to pursue.

Direct Simplification of a Given Model to a Reduced-order Form

Our approach to the realization problem, as discussed in Chapter 3 as well as in the foregoing part of this section, has been predicated on the assumption that the multiscale model is to be built from known covariance data. On the other hand, there may be circumstances in which we already have available a multiscale model, and our goal is to reduce this model to a simpler form. This new problem could be cast in the form of our standard problem; in

particular, we could derive the covariance structure of the given model, discard knowledge of the given model, and then proceed as we did in Chapter 3. However, we would intuitively expect that a more straightforward, effective approach would be to operate directly on the given multiscale model. We may obtain useful guidance by the time-series analogue of this problem, as described in [8, 20], with the analytical tools described therein being brought to bear in our context.

Identification of Structured Multiscale Processes

A problem that has recently received considerable attention in the literature is the direct identification of state-space models for stationary stochastic processes, indexed by discrete time [48,54]. In this problem, a state-space model is built *directly* from input-output data (or just from output data), without the intermediate step of estimating covariance or transfer functions; in this sense, these modern approaches are more appropriate for the handling of real data than the approach taken in Akaike's landmark paper [2].

These approaches are not appropriate for addressing the stochastic realization problem that we have studied in Chapter 3. One of the principal difficulties is the identification schemes in [48, 54] assume and exploit the considerable structure that is contained in stationary stochastic processes indexed by discrete time. On the other hand, there may be some merit in focusing our attention on a restricted class of multiscale stochastic processes that have a special, regular structure in scale, analogous to the structure in time of stationary time series. For such a restricted multiscale class, the theory and techniques developed in [48, 54] might be applicable in suitably modified form.

We now illustrate the type of special, regular structure that might be required of a multiscale process, in order to render it amenable to the time-series techniques. As a caveat, we remark that this description is purely speculative; we have not yet done any analysis of this problem. Proceeding, consider a multiscale process in which the interpolation matrices $A(s)$ are constant across all scales, and the noise-shaping matrices $B(s)$ are also constant, modulo a scale-factor:

$$A(s) \;=\; A \qquad \text{and} \qquad B(s) \;=\; 2^{-\mu m(s)} B.$$

If we somehow knew the scale factor μ, we could perhaps invert its effect, and thereby cast the problem in a form that is quite close to the time-series identification problem.

Explicit embedding of coarse-scale information

Throughout Chapter 3, we restricted attention to a multiscale realization problem in which we explicitly constrained only the finest-scale correlation structure. In some applications, we may also wish to constrain, at least partially, the coarse-scale correlation structure. For example, we may wish to reconstruct the value of some process, based on sensor measurements that are best modeled as noisy, non-local versions of the finest-scale process. In this situation, it is reasonable to model explicitly this non-local information as a coarse-scale process state.

To address this more general modeling problem, we should certainly, if possible, take maximal advantage of the tools developed in Chapter 3. A most straightforward way to proceed is to first devise matrices W_s exactly as in Sections 3.4, without regard for the explicit coarse-scale state information we wish to embed. Then, as a second step, we could augment these W_s matrices with additional rows that capture the desired coarse scale

information; for instance, if we wish to have an average of the whole process as a state, then we could augment W_0 with a rows consisting of all ones. Finally, using these augmented W_s matrices, we could find parameter values for $P(0)$, $A(s)$ and $B(s)$ using (3.5), (3.9) and (3.10), respectively.

One difficulty with the approach just described is that the important coarse-scale state information does not necessarily propagate to finer scales in the consistent manner we would like; this problem was discussed in Section 3.7.2. As an alternative, one could use a modified version of the realization approach described in Section 3.8; this may hold promise, since the propagation of information is handled more explicitly. The challenge, though, remains to streamline computationally the approach of Section 3.8.

Miscellaneous other issues

Our choice for the information content of states in multiscale processes has been heavily influenced by canonical correlation analysis; there would certainly be some benefit in exploring other methodologies for constructing the process states. In fact, it is argued in [64] that the normalization involved in the definition of both the correlation coefficient $\rho(\cdot)$ and its generalization $\bar{\rho}(\cdot)$ can destroy important information concerning the energy (i.e., variance) of the correlated components in a pair of random vectors. The authors of [64] thus argue that the canonical correlation coefficients can be misleading, and that an alternative, superior metric can be devised. As a second alternative, the authors of [8] consider a time-series stochastic realization problem in which the state is chosen to minimize the covariance of a prediction of the future, based on a linear function of the past.

Proposition 1 provides a lower bound on the state dimension required for an exact realization of prespecifed finest-scale statistics. However, as revealed by our construction of a multiscale model for Brownian motion (see Section 2.3), this bound is not tight, even if we restrict attention to processes indexed on the dyadic tree. Thus, a remaining open question is to devise a tighter bound for exact realizations.

Our use of the selection matrices Θ_s and Θ_{s^c} (in Section 3.4.3) to calculate canonical correlation matrices leads to approximate results in the non-WSMRF case. An open issue is to quantify the fidelity of this approximation, as a function of the window width implied by the dimension of the Θ_s and Θ_{s^c} matrices.

Finally, there may be benefit in utilizing matrices Θ_s and Θ_{s^c} that capture more than just boundary information. For instance, if the selection-matrix structure were relaxed, then we could envision the components of, say, $\Theta_{s^c}\chi_{s^c}$ containing both spatially localized, fine-scale information as well as more spatially distributed, coarse-scale information.

6.2.2 A Multiresolution Approach to Discriminating Targets from Clutter in SAR Imagery

Our results in Chapter 4 suggest a number of interesting possibilities to pursue in future work. For one, there are connections between the multiresolution approach taken here and the one taken in [61]; the two approaches have the same spirit, but are different in some important ways, and there may be merit in clarifying the connections between the two. For example, theoretical models are developed in [61] that could possibly be used to either validate or refine the multiresolution models we developed here. Furthermore, certain optimal resolutions are identified and exploited in [61]; perhaps we could use these optimal scales in lieu of the dyadic progression of scales used here.

A second possible extension of the work here would be to develop more sophisticated multiscale models. Recall, for example, the residuals in Figure 4-6 resulting from application of our man-made model to a multiresolution image sequence containing a man-made object. These residuals demonstrate that the man-made model is not completely capturing the scale-to-scale statistical coupling of the images; there is a need for better accounting of the dominant scatterers in the images. Perhaps this need could be fulfilled by developing a whole collection of man-made models, with each individual model specialized to a particular target configuration. Given this collection of models, we could then envision carrying out likelihood-based target recognition. We emphasize that the efficiency of our likelihood calculations would be instrumental in the practical applicability of this approach to recognition.

Another possible extension would be to exploit the multiresolution models to carry out image compression, in a manner analogous to the use of linear predictive coding for time-series (e.g., speech) compression. Finally, there is the possibility of developing multiscale models for remote sensing applications, such as classification of terrain cover, for which we could develop a number of natural-clutter models, including one for trees, another for grass, and so forth.

In general, the multiresolution nature of both our statistical models and our algorithms provide a very natural way for managing the considerable computational burden of this detection problem, especially in the likely scenario that there is a large amount of data, representing wide area surveillance.

6.2.3 An Overlapping-Tree Approach to Modeling and Estimation

In Figure 5-7, we displayed sample paths of the wood texture, generated using both non-overlapping and overlapping tree processes. Qualitatively, the sample path generated with the overlapping model (i.e., the sample path in (c)) looks more like the actual wood texture (shown in (a)) than the sample path generated with the non-overlapping model (i.e., the sample path in (b)). However, according to the more quantitative measure of correlation-matching, we find in Figure 5-8 that actually the non-overlapping model is superior. In light of these results, a natural question is whether the overlapping model could be improved by increasing the amount of overlap in the vertical direction.

In an unrelated vein, we have not explored the full variety of possibilities in specifying the values of H_x and R_l. With regard to the former, we exclusively used in our numerical experiments a linear tapering for H_x; the structure of this linear tapering was described in Section 5.5. Certainly, however, there exist other possible tapering schemes, such as a quadratic one. With regard to R_l, we exclusively used the rule (5.27) in our numerical experiments, although (5.26) allows for considerably greater flexibility. By intelligently taking advantage of this flexibility, we may be able to further reduce the MSE of our overlapping-tree estimator.

Appendix A

Proof of Propositions 2, 3 and 4

A.1 Proof of Proposition 2

Let η_1 and η_2 be random vectors of dimension n_1 and n_2. Let P_{η_i} be the covariance of η_i, for $i = 1, 2$ and let $P_{\eta_1\eta_2}$ be the cross-covariance between η_1 and η_2. Finally, suppose that P_{η_i} has rank m_i, for $i = 1, 2$.

Proposition 2: *There exist matrices T_1 and T_2, of dimension $m_1 \times n_1$ and $m_2 \times n_2$, respectively, such that*

$$\begin{pmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{pmatrix} \begin{pmatrix} P_{\eta_1} & P_{\eta_1\eta_2} \\ P_{\eta_1\eta_2}^T & P_{\eta_2} \end{pmatrix} \begin{pmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{pmatrix}^T = \begin{pmatrix} I_{m_1} & D \\ D^T & I_{m_2} \end{pmatrix}, \tag{A.1}$$

and

$$\begin{pmatrix} T_1^+ & \mathbf{0} \\ \mathbf{0} & T_2^+ \end{pmatrix} \begin{pmatrix} I_{m_1} & D \\ D^T & I_{m_2} \end{pmatrix} \begin{pmatrix} T_1^+ & \mathbf{0} \\ \mathbf{0} & T_2^+ \end{pmatrix}^T = \begin{pmatrix} P_{\eta_1} & P_{\eta_1\eta_2} \\ P_{\eta_1\eta_2}^T & P_{\eta_2} \end{pmatrix}. \tag{A.2}$$

In these equations, I_{m_i} is an identity matrix of dimension $m_i \times m_i$ (for $i = 1, 2$). The matrix D has dimension $m_1 \times m_2$ and is given by

$$D = \begin{pmatrix} \hat{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{A.3}$$

where $\hat{D}$ is a positive definite diagonal matrix given by

$$\hat{D} = diag(d_1, d_2, ..., d_{m_{12}}), \quad 1 \geq d_1 \geq d_2 \geq ... \geq d_{m_{12}} > 0. \tag{A.4}$$

Finally, T_i^+ is the Moore-Penrose pseudoinverse of T_i, and is given by

$$T_i^+ = P_{\eta_i} T_i^T, \qquad (i = 1, 2). \tag{A.5}$$

Proof: We prove the proposition by construction. Our strategy is to construct T_1 and T_2 in a sequence of two steps. In the first step, we devise a pair of *whitening* matrices W_1 and W_2, defined such that for $\eta_{w_i} = W_i \eta_i$ (for $i = 1, 2$) we have

$$E[\eta_{w_i} \eta_{w_i}^T] = W_i P_{\eta_i} W_i^T$$

$$= I_{m_i}, \qquad (i = 1, 2). \tag{A.6}$$

In the second step, we devise a second pair of transformation matrices $\hat{W}_1$ and $\hat{W}_2$, defined such that with $\eta_{\hat{w}_i} = \hat{W}_i \eta_{w_i}$ (for $i = 1, 2$), we retain the whiteness of η_{w_i}, and additionally, we render diagonal the cross-covariance between $\eta_{\hat{w}_1}$ and $\eta_{\hat{w}_2}$:

$$\begin{aligned} E[\eta_{\hat{w}_1} \eta_{\hat{w}_2}^T] &= \hat{W}_1 W_1 P_{\eta_1 \eta_2} W_2^T \hat{W}_2^T \\ &= D. \end{aligned} \tag{A.7}$$

Finally, we satisfy (A.1), by defining T_i as follows:

$$T_i = \hat{W}_i W_i, \qquad (i = 1, 2). \tag{A.8}$$

To construct W_1 and W_2, we first let $\lambda_{i,2}, \ldots, \lambda_{i,m_i}$ be the nonzero eigenvalues of P_{η_i}, and let $S_{i,1}$, $S_{i,2}, \ldots S_{i,m_i}$ be the corresponding eigenvectors. Consolidating this eigendecomposition information into the eigenvalue and eigenvector matrices Λ_i, S_i,

$$\Lambda_i = \operatorname{diag}(\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,m_i}), \tag{A.9}$$

$$S_i = \begin{pmatrix} S_{i,1} & | & S_{i,2} & | & \cdots & | & S_{i,m_i} \end{pmatrix} \qquad (i = 1, 2). \tag{A.10}$$

we have that

$$\begin{aligned} P_{\eta_i} &= S_i \Lambda_i S_i^T, \quad i = 1, 2, \\ S_i^T S_i &= I_{m_i}. \end{aligned}$$

We then define W_i, for $i = 1, 2$, as follows:

$$W_i = \Lambda_i^{-1/2} S_i^T, \quad i = 1, 2.$$

which has the desired whitening property (A.6).

To construct $\hat{W}_i$, we first do a singular value decomposition of the cross correlation matrix for η_{w_1} and η_{w_2}:

$$\begin{aligned} E(\eta_{w_1} \eta_{w_2}^T) &= W_1 P_{\eta_1 \eta_2} W_2^T \\ &= U_1 D U_2^T, \end{aligned}$$

where U_i is an $m_i \times m_i$ orthonormal matrix ($i = 1, 2$), and D is the diagonal matrix of singular values. In terms of this decomposition, we readily verify that by letting

$$\hat{W}_i = U_i^T, \quad i = 1, 2,$$

the property (A.7) is fulfilled, and hence this is how we define $\hat{W}_i$. Finally, we define T_1 and T_2 as in (A.8), for which (A.1) is satisfied.

We now verify, in turn, (A.5) and (A.2). By construction, T_1 and T_2 have full row rank. Hence, the Moore-Penrose pseudoinverse of T_i must be

$$\begin{aligned} T_i^+ &= T_i^T \left(T_i T_i^T \right)^{-1} \\ &= S_i \Lambda_i^{-1/2} U_i (U_i^T \Lambda_i^{-1/2} S_i^T S_i \Lambda_i^{-1/2} U_i)^{-1} \end{aligned}$$

$$
\begin{aligned}
&= S_i \Lambda_i^{1/2} U_i \\
&= P_{\eta_i} T_i^T \qquad (i = 1, 2).
\end{aligned}
$$

thus verifying (A.5). Using this definition for T_i^+, we verify (A.2) by straightforward calculation:

$$
\begin{aligned}
T_i^+ \left(T_i^+\right)^T &= \left(S_i \Lambda_i^{1/2} U_i\right) \left(U_i^T \Lambda_i^{1/2} S_i^T\right) \\
&= P_{\eta_i} \qquad (i = 1, 2) \\
T_1^+ D \left(T_2^+\right)^T &= S_1 \Lambda_1^{1/2} (U_1 D U_2^T) \Lambda_2^{1/2} S_2^T \\
&= S_1 \Lambda^{1/2} (\Lambda_1^{-1/2} S_1^T P_{\eta_1 \eta_2} S_2 \Lambda_2^{-1/2}) \Lambda_2^{1/2} S_2^T \\
&= S_1 S_1^T P_{\eta_1 \eta_2} S_2 S_2^T \\
&= P_{\eta_1 \eta_2}.
\end{aligned}
$$

Finally, we demonstrate that $\text{rank}(D) = \text{rank}(P_{\eta_1 \eta_2})$. Since $D = T_1 P_{\eta_1 \eta_2} T_2^T$, we see that $\text{rank}(D) \leq \text{rank}(P_{\eta_1 \eta_2})$; on the other hand, since $P_{\eta_1 \eta_2} = T_1^+ D (T_2^+)^T$, we see that $\text{rank}(P_{\eta_1 \eta_2}) \leq \text{rank}(D)$. Hence, the equality of the ranks of D and $P_{\eta_1 \eta_2}$ is established, and the proof is complete. **QED**.

A.2 Proof of Proposition 3

Proposition 3 *Let W_1 and W_2 be matrices of dimension $m_1 \times n_1$ and $m_2 \times n_2$, respectively, such that*

$$
W_i P_{\eta_i} W_i^T = I_{m_i} \qquad (i = 1, 2)
$$

Then, for all such W_1 and W_2, the nonzero singular values of $W_1 P_{\eta_1 \eta_2} W_2^T$ are given by the diagonal entries of the matrix $\hat{D}$, which is unique.

Proof: We continue to use the matrices S_i and Λ_i, for $i = 1, 2$, as defined in (A.9) and (A.10). Now, in light of our assumption concerning W_1 and W_2, we see that

$$
W_i \left(S_i \Lambda_i S_i^T\right) W_i^T = I_{m_i} \qquad (i = 1, 2),
$$

which means that $W_1 S_1 \Lambda_1^{1/2}$ and $W_2 S_2 \Lambda_2^{1/2}$ are unitary. Using this fact, we can characterize the singular values of $W_1 P_{\eta_1 \eta_2} W_2^T$ in the following way. We note by straightforward calculation that

$$
\begin{aligned}
W_1 P_{\eta_1 \eta_2} W_2^T &= W_1 \left(S_1 S_1^T P_{\eta_1 \eta_2} S_2 S_2^T\right) W_2^T \\
&= \left(W_1 S_1 \Lambda_1^{1/2}\right) \left(\Lambda_1^{-1/2} S_1^T P_{\eta_1 \eta_2} S_2 \Lambda_2^{-1/2}\right) \left(\Lambda_2^{1/2} S_2^T W_2^T\right) \\
&= \left(W_1 S_1 \Lambda_1^{1/2}\right) \left(U_1 D U_2^T\right) \left(\Lambda_2^{1/2} S_2^T W_2^T\right).
\end{aligned}
$$

But, we can pre-multiply and post-multiply any given matrix by a unitary matrices, with no effect on the non-zero singular values of the given matrix. Hence, the singular values of $W_1 P_{\eta_1 \eta_2} W_2^T$ are given by the diagonal elements of D, which coincide with the diagonal elements of $\hat{D}$. **QED**.

A.3 Proof of Proposition 4

Proposition 4 *Let $(\hat{T}_1, \hat{T}_2, \hat{D})$ be the canonical correlation matrices for (μ_1, μ_2). If (μ_1, μ_2) are related to (η_1, η_2), as in (2.38) and (2.39), then $(\hat{T}_1\Theta_1, \hat{T}_2\Theta_2, \hat{D})$ are the canonical correlation matrices for (η_1, η_2).*

We begin by establishing that (η_1, η_2) and (μ_1, μ_2) share the same number of nonzero canonical correlation coefficients; by Proposition 2, this result will follow, if we can establish that $E(\eta_1\eta_2^T)$ and $E(\mu_1\mu_2^T)$ share the same rank. But, $E(\mu_1\mu_2^T) = \Theta_1 E(\eta_1\eta_2^T)\Theta_2^T$ and hence $\text{rank}[E(\mu_1\mu_2^T)] \leq \text{rank}[E(\eta_1\eta_2^T)]$. On the other hand (2.38) implies that there exist matrices F_1 and F_2 such that

$$\eta_i \;=\; F_i\mu_i \,+\, \tilde{\eta}_i \qquad (i = 1, 2),$$

where $\tilde{\eta}_1$ and $\tilde{\eta}_2$ are independent, and each is independent of both μ_1 and μ_2. Therefore, $E(\eta_1\eta_2^T) = F_1 E(\mu_1\mu_2^T)F_2^T$, and hence $\text{rank}[E(\eta_1\eta_2^T)] \leq \text{rank}[E(\mu_1\mu_2^T)]$. Combining these inequalities, we conclude that (η_1, η_2) and (μ_1, μ_2) share the same number of nonzero canonical correlation coefficients.

Now, by assumption,

$$\begin{pmatrix} \hat{T}_1 & \mathbf{0} \\ \mathbf{0} & \hat{T}_2 \end{pmatrix} \begin{pmatrix} \Theta_1 P_{\eta_1} \Theta_1^T & \Theta_1 P_{\eta_1\eta_2} \Theta_2^T \\ \Theta_2 P_{\eta_1\eta_2}^T \Theta_1^T & \Theta_2 P_{\eta_2} \Theta_2^T \end{pmatrix} \begin{pmatrix} \hat{T}_1 & \mathbf{0} \\ \mathbf{0} & \hat{T}_2 \end{pmatrix}^T = \begin{pmatrix} I & \hat{D} \\ \hat{D} & I \end{pmatrix}.$$

Hence, by straightforward algebraic rearrangement,

$$\begin{pmatrix} \hat{T}_1\Theta_1 & \mathbf{0} \\ \mathbf{0} & \hat{T}_2\Theta_2 \end{pmatrix} \begin{pmatrix} P_{\eta_1} & P_{\eta_1\eta_2} \\ P_{\eta_1\eta_2}^T & P_{\eta_2} \end{pmatrix} \begin{pmatrix} \hat{T}_1 & \mathbf{0} \\ \mathbf{0} & \hat{T}_2 \end{pmatrix}^T = \begin{pmatrix} I & \hat{D} \\ \hat{D} & I \end{pmatrix}.$$

Appealing, finally, to the uniqueness property of $\hat{D}$, as asserted by Proposition 2, the proposition follows. **QED**.

Appendix B

Proof of Propositions 5 and 6

In this appendix, we prove Propositions 5 and 6. As in the main text, we let η, η_1 and η_2 be random vectors, having covariance matrices P_η, P_{η_1} and P_{η_2} respectively, exactly as in Section 2.4 (see (2.24) and (2.25)). Also, we let $(\hat{T}_1, \hat{T}_2, \hat{D})$ be the canonical correlation matrices associated with (η_1, η_2), with $\hat{D} = \mathrm{diag}(d_1, d_2, \ldots, d_{m_{12}})$.

To review, here are the results we will prove:

Proposition 5 *For $i = 1, 2$ and for all matrices W_i,*

$$\bar{\rho}(\eta_1, \eta_2 \mid W_i \eta_i) \;\leq\; \bar{\rho}(\eta_1, \eta_2).$$

Proposition 6 *For $1 \leq k < m_{12}$ and for $i = 1, 2$,*

$$\min_{W \in \mathcal{M}_k} \bar{\rho}(\eta_1, \eta_2 \mid W\eta) \;=\; \min_{W_i \in \mathcal{M}_k} \bar{\rho}(\eta_1, \eta_2 \mid W_i\eta_i) \;=\; \bar{\rho}(\eta_1, \eta_2 \mid \hat{T}_{i,k}\eta_i) \;=\; d_{k+1}.$$

For $k \geq m_{12}$,

$$\min_{W \in \mathcal{M}_k} \bar{\rho}(\eta_1, \eta_2 \mid W\eta) \;=\; \min_{W_i \in \mathcal{M}_k} \bar{\rho}(\eta_1, \eta_2 \mid W_i\eta_i) \;=\; \bar{\rho}(\eta_1, \eta_2 \mid \hat{T}_i\eta_i) \;=\; 0.$$

Throughout the proofs, we assume without loss of generality that

$$\Sigma \;=\; E\left[\begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} \begin{pmatrix} \eta_1^T & \eta_2^T \end{pmatrix}\right] \;=\; \begin{pmatrix} I_{n_1} & D \\ D^T & I_{n_2} \end{pmatrix}. \tag{B.1}$$

There is no loss, thanks to the invertibility of the transformation in (2.26) and (2.27), which implies that we can reduce the case of arbitrary covariance for $\begin{pmatrix} \eta_1^T & \eta_2^T \end{pmatrix}^T$ to form in (B.1) by exploiting the identity

$$\min_{W \in \mathcal{M}_k} \bar{\rho}(\eta_1, \eta_2 \mid W\eta) \;=\; \min_{W_i \in \mathcal{M}_k} \bar{\rho}(T_1\eta_1, T_2\eta_2 \mid WT\eta), \tag{B.2}$$

where T is the following block-diagonal matrix:

$$T \;\equiv\; \mathrm{diag}\,(T_1, T_2)\,.$$

B.1 A Useful Lemma

The following lemma will be instrumental to both proofs, as it relates $\bar{\rho}(\eta_1, \eta_2)$ to $\bar{\rho}(\eta_1, \eta_2 \mid W_1\eta_1)$ in terms of the matrix D.

Lemma 2 *Let $W_1 \in \mathcal{R}^{k \times n_1}$, for any fixed k, with W_1 having orthonormal rows. Let $W_1^\perp$ be a matrix whose rows form an orthonormal basis for the nullspace of W_1. Then,*

$$\bar{\rho}(\eta_1, \eta_2 \mid W_1\eta_1) = \max_{f_1 \in F_1,\, f_2 \in F_2} \left\{ f_1^T W_1^\perp D f_2 \right\} \tag{B.3}$$

$$= \max_{f_2 \in F_2} \| W_1^\perp D f_2 \|_2, \tag{B.4}$$

where F_1 and F_2 denote the following sets:

$$F_1 = \left\{ f \in \mathcal{R}^{n_1 - k};\ f^T f = 1 \right\},$$
$$F_2 = \left\{ f \in \mathcal{R}^{n_2};\ f^T (I - D^T W_1^T W_1 D) f = 1 \right\}. \tag{B.5}$$

Proof: For any unitary matrix U,

$$\bar{\rho}(\eta_1, \eta_2 \mid W_1\eta_1) = \bar{\rho}(U^T\eta_1, \eta_2 \mid (W_1U)(U^T\eta_1)). \tag{B.6}$$

Letting $U \equiv \begin{pmatrix} W_1^T & (W_1^\perp)^T \end{pmatrix}$, we see that the conditional covariance of $\begin{pmatrix} (U^T\eta_1)^T & \eta_2^T \end{pmatrix}^T$ is given by

$$\mathrm{cov}\left[\begin{pmatrix} U^T\eta_1 \\ \eta_2 \end{pmatrix} \mid (W_1U)(U^T\eta_1) \right] = \begin{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n_1-k} \end{pmatrix} & \begin{pmatrix} \mathbf{0} \\ W_1^\perp D \end{pmatrix} \\ \begin{pmatrix} \mathbf{0} \\ W_1^\perp D \end{pmatrix}^T & I - D^T W_1^T W_1 D \end{pmatrix}.$$

Hence, adapting (3.15)-(3.16) to this special case, the lemma follows. **QED**.

B.2 Proof of Proposition 5

We know from (3.13) that

$$\bar{\rho}(\eta_1, \eta_2) = d_1.$$

Combining this fact with (B.4), it follows that the Proposition will be proved if we can show that

$$\max_{f_2 \in F_2} \| W_1^\perp D f_2 \|_2^2 \leq d_1^2. \tag{B.7}$$

To establish (B.7), we first note that since the rows of $W_1^\perp$ form an orthonormal basis for the row space of W_1, we have that $\forall x$,

$$\| x \|_2^2 = \| W_1 x \|_2^2 + \| W_1^\perp x \|_2^2. \tag{B.8}$$

Since $\forall f_2 \in F_2$,

$$\begin{aligned} f_2^T(I - D^T W_1^T W_1 D) f_2 &= \| f_2 \|_2^2 - \| W_1 D f_2 \|_2^2 \\ &= 1, \end{aligned} \tag{B.9}$$

we can apply (B.8) in (B.9) with $x = Df_2$ to see that $\forall f_2 \in F_2$,

$$\| W_1^{\perp} D f_2 \|_2^2 = \| Df_2 \|_2^2 - \| f_2 \|_2^2 + 1, \quad \forall f_2 \in F_2. \tag{B.10}$$

But

$$\begin{aligned} \min_{f_2 \in F_2} \left\{ \| f_2 \|_2^2 - \| Df_2 \|_2^2 \right\} &= \min_{f_2 \neq 0} \left\{ \frac{f_2^T \left(I - D^T D \right) f_2}{f_2^T \left(I - D^T W_1^T W_1 D \right) f_2} \right\} \\ &\geq \min_{f_2 \neq 0} \left\{ \frac{f_2^T (I - D^T D) f_2}{f_2^T f_2} \right\} \min_{f_2 \neq 0} \left\{ \frac{f_2^T f_2}{f_2^T (I - D^T W_1^T W_1 D) f_2} \right\} \\ &= \lambda_{min} \left(I - D^T D \right) \lambda_{min} \left[\left(I - D^T W_1^T W_1 D \right)^{-1} \right] \\ &= (1 - d_1^2)\,(1) \end{aligned} \tag{B.11}$$

where $\lambda_{min}(\cdot)$ denotes the smallest eigenvalue of the enclosed matrix expression. In the third line, we have used Rayleigh's principle [60], which asserts that for any pair of symmetric, positive definite matrices A and B,

$$\min_{x \neq 0} \frac{x^T B x}{x^T A x} = \lambda_{min}(A^{-1} B).$$

By combining (B.10) and (B.11), the desired result (B.7) is established. **QED**.

B.3 Proof of Proposition 6

For the purposes of establishing Proposition 6, we not only assume that (B.1) holds, but also that all elements of the matrix D have values strictly less than one. If, to the contrary, the first k_1 diagonal elements of D were equal to one, then k would *have* to be at least as great as k_1 in order to reduce $\bar{\rho}(\eta_1, \eta_2 \mid W\eta)$ below unity. Furthermore, we could, without loss of generality or optimality, let the first k_1 rows of W be $\begin{pmatrix} I_{k_1} & \mathbf{0} \end{pmatrix}$; expressing, then, W as

$$W = \begin{pmatrix} I_k & \mathbf{0} \\ W_{remain} \end{pmatrix},$$

we could solve for W_{remain} (having $k - k_1$ rows) as

$$\arg \min_{W \in \mathcal{M}_{k-k_1}} \bar{\rho}(\eta_1, \eta_2 \mid \begin{pmatrix} I_{k_1} & \mathbf{0} \end{pmatrix} \eta, W_{remain} \eta).$$

Also, we restrict attention throughout our proof to matrices $W \in \mathcal{M}_k$ having full row rank.

B.3.1 Proof of Proposition 6

We begin by temporarily constraining W to have either of the two forms

$$W = \begin{pmatrix} W_1 & \mathbf{0} \end{pmatrix} \text{ or } \begin{pmatrix} \mathbf{0} & W_2 \end{pmatrix}, \tag{B.12}$$

and we find a matrix W that minimizes $\bar{\rho}(\eta_1, \eta_2 \mid W\eta)$, subject to this additional constraint. The following Lemma summarizes the key result here, with the proof contained in Section B.3.2.

Lemma 3

$$\begin{aligned} \min_{W_1 \in \mathcal{M}_k} \bar{\rho}(\eta_1, \eta_2 \mid W_1\eta_1) &= \bar{\rho}(\eta_1, \eta_2 \mid \begin{pmatrix} I_k & \mathbf{0} \end{pmatrix} \eta_1) \\ &= \bar{\rho}(\eta_1, \eta_2 \mid \begin{pmatrix} I_k & \mathbf{0} \end{pmatrix} \eta_2) \\ &= \min_{W_2 \in \mathcal{M}_k} \bar{\rho}(\eta_1, \eta_2 \mid W_2\eta_2) \\ &= d_{k+1}. \end{aligned}$$

Next, we establish that in fact there is no loss of optimality in the additional constraint in (B.12). The following lemma summarizes the key result here, with the proof also contained in Section B.3.2.

Lemma 4 *For any matrix $W \in \mathcal{R}^{k \times (n_1+n_2)}$ having full row rank, there exists a pair of matrices $W_1 \in \mathcal{R}^{k_1 \times n_1}$ and $W_2 \in \mathcal{R}^{k_2 \times n_2}$, with $k_1 + k_2 \leq k$, such that*

$$\bar{\rho}(\eta_1, \eta_2 \mid W_1\eta_1, W_2\eta_2) \leq \bar{\rho}(\eta_1, \eta_2 \mid W\eta).$$

To make clear the consequences of Lemma 3, let us suppose that

$$W^* \in \mathcal{M}_k \text{ minimizes } \bar{\rho}(\eta_1, \eta_2 \mid W\eta).$$

Then, from Lemma 3, we know there exist matrices $W_1^* \in \mathcal{M}_{k_1}$ and $W_2^* \in \mathcal{M}_{k_2}$ (for some k_1 and k_2 such that $k_1 + k_2 = k$) such that

$$\bar{\rho}(\eta_1, \eta_2 \mid W^*\eta) = \bar{\rho}(\eta_1, \eta_2 \mid W_1^*\eta_1, W_2^*\eta_2).$$

Then, fixing W_1^*, let us define

$$\tilde{\eta}_i \equiv \eta_i - E(\eta_i \mid W_1^*\eta_1), \quad (i = 1, 2),$$

which we use to see that

$$\begin{aligned} \bar{\rho}(\eta_1, \eta_2 \mid W^*\eta) &= \min_{W_2 \in \mathcal{M}_{k_2}} \bar{\rho}(\eta_1, \eta_2 \mid W_1^*\eta_1, W_2\eta_2) \\ &= \min_{W_2 \in \mathcal{M}_{k_2}} \bar{\rho}(\tilde{\eta}_1, \tilde{\eta}_2 \mid W_2\tilde{\eta}_2) \\ &= \min_{W_1 \in \mathcal{M}_{k_2}} \bar{\rho}(\tilde{\eta}_1, \tilde{\eta}_2 \mid \bar{W}_1\tilde{\eta}_1) \\ &= \min_{\bar{W}_1 \in \mathcal{M}_{k_2}} \bar{\rho}(\eta_1, \eta_2 \mid W_1^*\eta_1, \bar{W}_1\eta_1) \\ &= \min_{W_1 \in \mathcal{M}_k} \bar{\rho}(\eta_1, \eta_2 \mid W_1\eta_1). \end{aligned}$$

The first line follows directly from Lemma 3. The second line follows from the definition of $\tilde{\eta}_i$. The third line follows from Lemma 2. The fourth line follows again from the relation between $\tilde{\eta}_i$ and η_i. Finally, the fifth line follows from the fact that both of the vectors of conditioning information in the fourth line of functions only of η_1. With the exception of Lemmas 2 and 3, which follow, the proof of Proposition 6 is now complete. **QED**.

B.3.2 Proof of Lemmas 2 and 3

Proof of Lemma 2 In light of Lemma 1, and in particular (B.3), it is sufficient to devise particular values for $f_1 \in F_1$ and $f_2 \in F_2$ for which

$$f_1^T (W_1^\perp D) f_2 \;\geq\; d_{k+1}, \tag{B.13}$$

thus implying that

$$\bar{\rho}(\eta_1, \eta_2 \mid W_1 \eta_1) \geq d_{k+1},$$

where this bound must be tight, since by inspection,

$$\bar{\rho}(\eta_1, \eta_2 \mid \begin{pmatrix} I_k & \mathbf{0} \end{pmatrix} \eta_1) \;=\; d_{k+1}. \tag{B.14}$$

To establish (B.13), we first note that at least one of the unit vectors

$$e_1^T,\; e_2^T,\; \ldots,\; e_{k+1}^T$$

must belong to the row space of $W_1^\perp$, which itself has a dimension of $n_1 - k$; let us suppose that e_j^T belongs, with $\lambda^T W_1^\perp = e_j^T$ for some $\lambda \in \mathcal{R}^{n_1 - k}$. Now, exploiting the orthonormality of the rows of $W_1^\perp$, we see that $\lambda \in F_1$, and hence we let $f_1 = \lambda$. Also, we let $f_2 = e_j$, where the fact that

$$\begin{aligned} D e_j &= d_j e_j \\ &= d_j (W_1^\perp)^T f_1, \end{aligned}$$

implies that $W_1 D e_j = \mathbf{0}$, so that indeed $e_j \in F_2$. But for these values for f_1 and f_2,

$$\begin{aligned} f_1^T (W_1^\perp D) f_2 &= d_j \\ &\geq d_{k+1}, \end{aligned}$$

thus establishing (B.13) and completing the proof. **QED**.

Proof of Lemma 3 Let us express the matrix W in terms of its constituent rows as

$$W \;=\; \begin{pmatrix} W_1 & W_2 & \cdots & W_k \end{pmatrix}^T,$$

where the column vector W_i, for $i = 1, 2, \ldots, k$, can itself be decomposed as

$$W_i \;=\; \begin{pmatrix} W_{i,1} \\ W_{i,2} \end{pmatrix}, \quad W_{i,j} \in \mathcal{R}^{n_j},\; j = 1, 2.$$

Let us suppose that for some particular i, say i_1, $W_{i_1,1} \neq \mathbf{0}$, and $W_{i_1,2} \neq \mathbf{0}$. We demonstrate that W_{i_1} can be replaced with one of the two vectors V_1 or V_2, where

$$V_1 \equiv \begin{pmatrix} W_{i_1,1} \\ \mathbf{0} \end{pmatrix} \text{ and } V_2 \equiv \begin{pmatrix} \mathbf{0} \\ W_{i_1,2} \end{pmatrix},$$

with no incurred increase in the value of $\bar{\rho}(\eta_1, \eta_2 \mid W\eta)$. after the replacement.

At this point, we will find (3.13)-(3.16) quite useful. From (3.13), it immediately follows that

$$\tilde{P}_\eta W^T = \mathbf{0}. \tag{B.15}$$

which in turn implies that

$$\tilde{P}_\eta W_i = \mathbf{0},$$

so that

$$\begin{aligned} W_{i_1,1}^T \tilde{P}_{\eta_1} W_{i_1,1} &= W_{i_1,2}^T \tilde{P}_{\eta_2} W_{i_1,2} \\ &= -W_{i_1,1}^T \tilde{P}_{\eta_1\eta_2} W_{i_1,2}. \end{aligned}$$

There are now two possibilities:

$$W_{i_1,1}^T \tilde{P}_{\eta_1} W_{i_1,1} > 0, \quad \text{or} \quad W_{i_1,1}^T \tilde{P}_{\eta_1} W_{i_1,1} = 0.$$

The first implies that $\bar{\rho}(\eta_1, \eta_2 \mid Wx) = 1$, in which case there can certainly be no harm in our replacement strategy. The second implies that

$$\tilde{P}_{\eta_1} V_1 = \mathbf{0}, \quad \text{and} \quad \tilde{P}_{\eta_2} V_2 = \mathbf{0}, \tag{B.16}$$

which, in turn, means that there exist unique vectors λ_1 and λ_2 in $\mathcal{R}^k$ such that

$$V_i = W^T \lambda_i, \quad (i = 1, 2).$$

Now, by exhaustively considering the possibilities, one can verify that at least one of λ_1 and λ_2 must have a non-zero value in its i_1-th component, for otherwise, W could not have full row rank. If λ_1 (λ_2) has this property, then we can replace W_{i_1} with V_1 (V_2) with no change in the value of $\bar{\rho}(\eta_1, \eta_2 \mid Wx)$. **QED**.

Appendix C

Detection and Discrimination

C.1 Least-squares Estimation of Regression Coefficients

In Section 4.2.2, we described a least-squares procedure for estimating the regression coefficients in our models for SAR imagery. Central to that procedure was the need to solve (4.4) for the regression vector $\mathbf{a}_k$. Here we fulfill that need, obtaining an analytical expression for $\mathbf{a}_k$.

Our strategy is to recast the optimization problem in (4.4) as one of solving an overdetermined set linear equations in a least-squares sense. To proceed, we define the matrix J as

$$J \equiv \begin{pmatrix} I(s_1\bar{\gamma}) & I(s_1\bar{\gamma}^2) & \cdots & I(s_1\bar{\gamma}^R) \\ I(s_2\bar{\gamma}) & I(s_2\bar{\gamma}^2) & \cdots & I(s_2\bar{\gamma}^R) \\ \vdots & \vdots & \vdots & \vdots \\ I(s_N\bar{\gamma}) & I(s_N\bar{\gamma}^2) & \cdots & I(s_N\bar{\gamma}^R) \end{pmatrix},$$

where $s_1, s_2, \ldots, s_N$ denote the pixel locations in the image to be predicted at resolution k. We also define the vector b as

$$b \equiv \begin{pmatrix} I(s_1) \\ I(s_2) \\ \vdots \\ I(s_N) \end{pmatrix}.$$

In terms of J and b, we can recast the definition of $\mathbf{a}_k$ in (4.4) as

$$\mathbf{a}_k \equiv \arg\min_{\mathbf{a}_k \in \mathcal{R}^R} \left\{ (b - J\mathbf{a}_k)^T (b - J\mathbf{a}_k) \right\}. \tag{C.1}$$

Assuming that the columns of J are independent, which is virtually assured for all values of R of interest, the solution to (C.1) is well-known to be

$$\mathbf{a}_k = \left(J^T J \right)^{-1} J^T b.$$

C.2 Derivation of Expression for Multiresolution Discriminant

In Section 4.3.1, we described a straightforward procedure for calculating our multiresolution discriminant. Central to efficiency of that procedure was the decomposition (4.12) of each of the log-likelihood terms in our expression (4.11) for the discriminant. Here we establish the validity of that decomposition.

We continue to use the notation that was established in Section 4.3.1. Additionally, we define X_k to be a vector containing all the state vectors $x(s)$ at scale k, and similarly, we define Y_k to be a vector containing all the observations $y(s)$ at scale k.

In terms of these notational conventions, the multiresolution discriminant is given by

$$\log\left(P_{Y|H_1}(Y \mid H_1)\right) - \log\left(P_{Y|H_0}(Y \mid H_0)\right). \tag{C.2}$$

By elementary probability, we can factor each of the likelihoods in the following way:

$$\begin{aligned} P_{Y|H_i}(Y \mid \theta) &= P_{Y_{M_0},\ldots,Y_{M_1}|H_i}(Y_{M_0},\ldots,Y_{M_1} \mid H_i), \\ &= P_{Y_{M_0}|H_i}(Y_{M_0} \mid H_i) P_{Y_{M_0+1},\ldots,Y_{M_1}|Y_{M_0},H_i}(Y_{M_0+1},\ldots,Y_{M_1}|Y_{M_0},H_i) \\ &\vdots \\ &= P_{Y_{M_0}\mid\theta}(Y_{M_0} \mid H_i) \prod_{k=M_0+1}^{M_1} P_{Y_k|Y_{k-1},\ldots,Y_{M_0},H_i}(Y_k \mid Y_{k-1},\ldots,Y_{M_0},H_i). \end{aligned} \tag{C.3}$$

To simplify the right side of (C.3), we note that by construction (see Section 4.2.2) of our state-space models, the conditioning information contained in the set

$$\{Y_{k-1}, Y_{k-2}, \ldots, Y_{M_0}, H_i\}$$

is the equivalent to the conditioning information contained in the set

$$\{X_{k-1}, X_{k-2}, \ldots, X_{M_0}, H_i\}.$$

By combining this fact with the Markov properties of our multiscale models, we readily see that each element of the product on the right side of (C.3) can be factored as

$$\begin{aligned} P_{Y_k|Y_{k-1},\ldots,Y_{M_0},H_i}(Y_k \mid Y_{k-1},\ldots,Y_{M_0},H_i) &= P_{Y_k|X_{k-1},\ldots,X_{M_0},H_i}(Y_k \mid X_{k-1},\ldots,X_{M_0},H_i) \\ &= P_{Y_k|X_{k-1},H_i}(Y_k \mid X_{k-1},H_i), \end{aligned}$$

and thus,

$$P_{Y|H_i}(Y \mid H_i) = \prod_{k=M_0}^{M_1} P_{Y_k|X_{k-1},H_i}(Y_k \mid X_{k-1},H_i). \tag{C.4}$$

Again appealing to the Markov properties of our multiscale models, we can expand each of the elements in the product on the right side of (C.4) as

$$P_{Y_k|X_{k-1},H_i}(Y_k \mid X_{k-1},H_i) = \prod_{\{s;\, m(s)=k\}} P_{w(s)|H_i}\left(y(s) - \mathbf{a}_{k,H_i}^T x(s\bar{\gamma}) \mid H_i\right),$$

so that

$$P_{Y|H_i}(Y \mid H_i) = \prod_{k=M_0}^{M_1} \left[\prod_{\{s;\, m(s)=k\}} P_{w(s)|H_i}\left(y(s) - \mathbf{a}_{k,H_i}^T x(s\bar{\gamma}) \mid H_i\right) \right]. \qquad \text{(C.5)}$$

Finally, combining (C.2) with (C.5), we conclude that the multiresolution discriminant can be expressed as

$$\left\{ \sum_{k=M_0}^{M_1} \sum_{\{s;\, m(s)=k\}} \log\left[P_{w(s)|H_1}\left(y(s) - \mathbf{a}_{k,H_1}^T x(s\bar{\gamma})\right)\right] \right\} - \left\{ \sum_{k=M_0}^{M_1} \sum_{\{s;\, m(s)=k\}} \log\left[P_{w(s)|H_0}\left(y(s) - \mathbf{a}_{k,H_0}^T x(s\bar{\gamma})\right)\right] \right\},$$

which is in agreement with (4.12).

C.3 Description of Prescreening Algorithm

For the purposes of prescreening the SAR imagery, we use a two-parameter CFAR (constant false alarm rate) algorithm [28]. This algorithm can be viewed as an adaptive image-processing scheme that makes a decision at each pixel location indicating whether the pixel value belongs to a target distribution or a clutter distribution. Put simply, the algorithm takes advantage of the fact that in SAR imagery, targets typically appear brighter than non-targets.

To describe how the CFAR algorithm is implemented, we denote by x the value at the current pixel location or *test cell* under consideration. We assume that both the test cell value x and the cell values in the neighborhood of the test cell represent realizations of independent and identically distributed Gaussian random variables. Under these assumptions, we can estimate the parameters of the common clutter distribution by collecting a large number of neighboring pixel values and computing their sample mean $\hat{\mu}_c$ and standard deviation $\hat{\sigma}_c$. The two-parameter detector is then defined by the rule

$$\text{Declare} \left\{ \begin{array}{l} \text{target present} \\ \text{target absent} \end{array} \right\} \text{ if } \frac{x - \hat{\mu}_c}{\hat{\sigma}_c} \left\{ \begin{array}{l} > \\ \leq \end{array} \right\} K$$

where K is an algorithm parameter known as the *threshold* and where the ratio $(x - \hat{\mu}_c)/\hat{\sigma}_c$ is known as the *CFAR statistic*. In Figure C-1, we show the image stencil that is used for selecting the pixel locations used to estimate the mean and standard deviation of x. We note that a large guard area is used between the test cell and the outer-stencil cells so that the presence of a target does not affect the estimation of the clutter distribution parameters.

If the pixel values in the image are indeed independent, identically distributed and Gaussian distributed, and $\hat{\mu} = \mu$ and $\hat{\sigma} = \sigma$, then the detector just described will yield a constant false alarm rate; this rate will be a function of the threshold value K. If these assumptions do not hold, then the CFAR property is not guaranteed. Even in these cases, however, the above rule is often employed because it provides a reasonable algorithm for detecting targets in SAR imagery. In any case, we are compelled by tradition to continue to use the term CFAR to describe this algorithm.

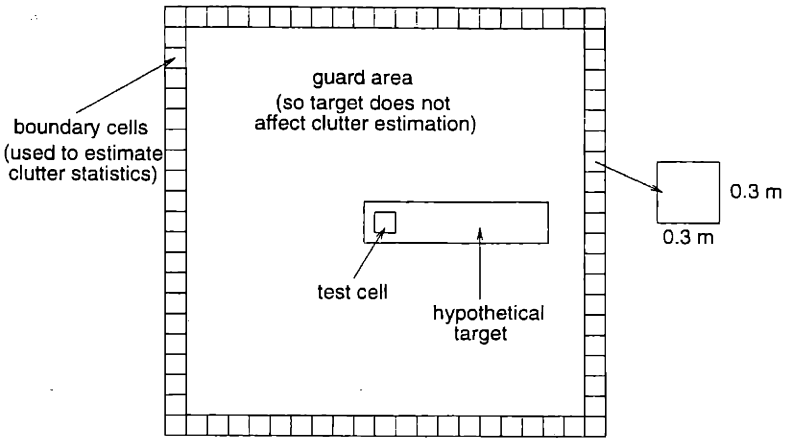


Figure C-1: Image stencil used to compute the target detection decision in the two-parameter CFAR algorithm. A large guard area is used so that the presence of a target does not affect the estimation of the clutter distribution statistics. The cells in the outer stencil are used to estimate the local mean and standard deviation of the clutter. If the value x in the test cell is at least K standard deviations beyond the mean, then a target is declared to be present.

By setting the threshold K to a suitably low value, we can ensure that at least one pixel will be assigned to the target class for every target of interest in the SAR imagery. Because we do not want to discard targets at the prescreening stage, we use such a threshold. In many instances, the CFAR algorithm will thus yield multiple detections on a single target, which we cluster together. Corresponding to each cluster, we create an ROI, with the centroid of the cluster aligned with the center of the ROI. In our application with 0.3m $\times$ 0.3m resolution imagery, each ROI consists of 128 $\times$ 128 pixels. Each ROI is passed on to the discriminator for further processing.

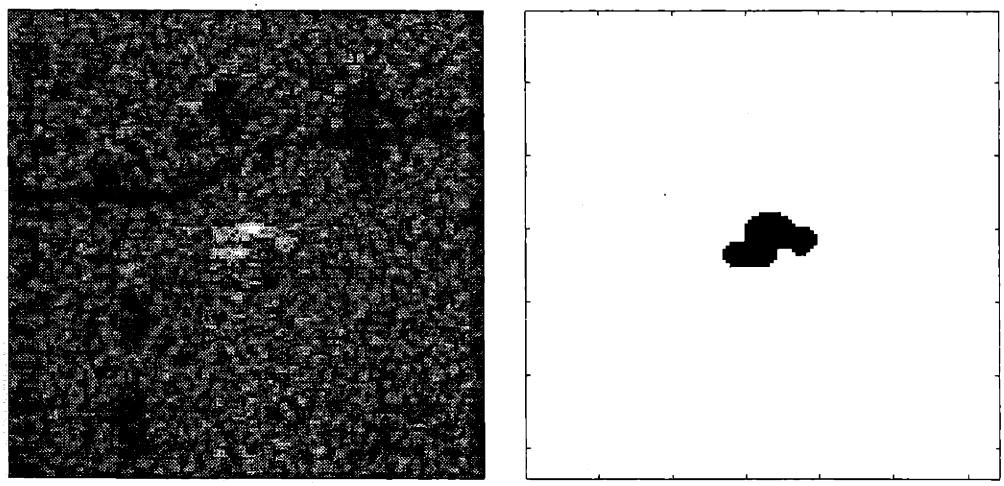
C.4 Description of Features in Lincoln Laboratory Discriminator

In this appendix, we provide a highly abridged version of the description in [40] of all the features used in the Lincoln Laboratory discriminator. These features are listed in Table 4.3.

C.4.1 Textural Features

We first consider the textural features *standard deviation*, *fractal dimension* and *ranked fill-ratio*. The values of these features are calculated using the pixel values in a special target-sized region within the ROI. This region is chosen in the following way: a target is hypothesized to exist in the ROI; then, the position and orientation of this hypothesized target are estimated by applying simple matched-filtering with a rectangular-shaped, target-sized template. We denote the resulting target-sized region by $\mathcal{T}$.

The *standard-deviation* feature is calculated as simply the sample standard deviation of the log-detected pixel values in the region $\mathcal{T}$. The *fractal-dimension* feature provides an estimate of the Hausdorff dimension of the spatial distribution of the brightest 50 pixel values in the region $\mathcal{T}$ [40]; the details of this feature's calculation are too involved to include here. The *rank-fill-ratio* feature is the percentage of power that is contained in the brightest five percent of the pixels in the region T.



(a) (b)

Figure C-2: Illustration of effect of morphological operations that are used to identify the principal object in an ROI. (a) Original ROI, containing a man-made object that appears target like. (b) Effect of morphological processing on ROI in (a).

C.4.2 Size Features

We now consider the size features *mass*, *diameter*, and *rotational inertia*. The values of these features are calculated using the pixel values in the region of the ROI containing the ROI's principal object. In a rough sense, this principal object is defined to be the bright blob near the center of the ROI. For example, in Figure C-2(a) we display an ROI containing a man-made object, and in Figure C-2(b), we display a binary-valued image that clearly identifies the location of the ROI's principal object. In general, the principal object in the ROI is found by applying morphological processing; we denote the resulting principal-object region by $\mathcal{P}$.

The *mass* feature is calculated as simply the number of pixels in the region $\mathcal{P}$. The *diameter* feature is equal to the length of the diagonal of the smallest rectangle (either horizontally oriented or vertically oriented) that encloses the region $\mathcal{P}$. The *rotational-inertia* feature is calculated as the second mechanical moment of the region $\mathcal{P}$ around its center of mass, normalized by the inertia of a square having equal mass.

C.4.3 Contrast features

Finally, we consider the contrast features *peak CFAR*, *mean CFAR* and *percent bright CFAR*. To calculate these features, we first apply the CFAR algorithm (see Appendix C.3) to a log-detected version of the ROI. In this way, we obtain a CFAR image, in which the pixel value at location (k,l) is given by

$$\frac{I(k,l) \; - \; \hat{\mu}_c}{\hat{\sigma}_c}$$

where $I(k,l)$ is the pixel value at location (k,l) in the log-detected image, and $\hat{\mu}_c$ and $\hat{\sigma}_c$ are estimates of the local mean and standard deviation, respectively, of the annular region surrounding pixel (k,l) as in Figure C-1.

The values of the contrast feature are calculated using the pixel values in the region $\mathcal{P}$

of the CFAR image. The *peak-CFAR* feature is simply the maximum value of the CFAR image, within the region $\mathcal{P}$. The *mean-CFAR* feature is the sample mean of the CFAR image within the region $\mathcal{P}$. The *percent-bright-CFAR* feature is the percentage of pixels in the region $\mathcal{P}$ of the CFAR image that exceed a certain CFAR value.

Appendix D

Proof of Proposition 7

In this appendix, we prove Proposition 7, which for convenience, we restate here.

Proposition 7 *Let χ be a random field with covariance P_χ and let $y = C\chi + v$ be a set of measurements with C a weighted selection matrix and R, the covariance of v, diagonal. Suppose we then choose G_x, H_x, G_y, C_l and R_l as described in Section 5.4.3. Then the optimal linear least-squares estimate $\hat{\chi}$ of χ based on y can either be computed directly or by lifting, performing optimal estimation in the lifted domain, and then projecting. That is, if $\hat{\chi} = Ly$, and $\hat{\chi}_l = L_l y_l$, then*

$$P_\chi C^T \left(CP_\chi C^T + R\right)^{-1} \;=\; L \;=\; H_x L_l G_y \;=\; H_x P_{\chi_l} C_l^T \left(C_l P_{\chi_l} C_l^T + R_l\right)^{-1} G_y$$

where P_{χ_l} is defined in (5.8). Moreover, if $P_{\tilde{\chi}}$ denotes the estimation error covariance in estimating χ based on y, and $P_{\tilde{\chi}_l}$ the estimation error covariance in estimating χ_l based on y_l, then

$$P_{\tilde{\chi}} = H_x P_{\tilde{\chi}_l} H_x^T$$

Our proof is facilitated by the following identity:

$$(CP_\chi C^T + R)^{-1} \;=\; G_y^T (C_l P_{\chi_l} C_l^T + R_l)^{-1} G_y \tag{D.1}$$

We prove (D.1) in the following way:

$$\begin{aligned}
&(CP_\chi C^T + R)[G_y^T (C_l P_{\chi_l} C_l^T + R_l)^{-1} G_y] \\
&\qquad = [(RG_y^T R_l^{-1} G_y) CP_\chi C^T + R][G_y^T (C_l P_{\chi_l} C_l^T + R_l)^{-1} G_y] \\
&\qquad = [RG_y^T R_l^{-1} C_l P_{\chi_l} C_l^T + RG_y^T](C_l P_{\chi_l} C_l^T + R_l)^{-1} G_y \\
&\qquad = [RG_y^T R_l^{-1} (C_l P_{\chi_l} C_l^T + R_l - R_l) + RG_y^T](C_l P_{\chi_l} C_l^T + R_l)^{-1} G_y \\
&\qquad = RG_y^T R_l^{-1} (C_l P_{\chi_l} C_l^T + R_l)(C_l P_{\chi_l} C_l^T + R_l)^{-1} G_y \\
&\qquad = I.
\end{aligned}$$

In the first line, we have exploited (5.26). Then, in the second line, we have used (5.23) and (5.8). In the third line, we have simply added and subtracted R_l, while in the fourth line, we have simply done a rearrangement of terms, which leads to some cancellation. Finally, in the fifth line, we have again used (5.26).

We now verify (5.28). We have the following sequence of identities:

$$\begin{aligned} L &= P_\chi C^T (C P_\chi C^T + R)^{-1} \\ &= P_\chi C^T G_y^T \left(C_l P_{\chi l} C_l^T + R_l \right)^{-1} G_y \\ &= P_\chi G_x^T C_l^T \left(C_l P_{\chi l} C_l^T + R_l \right)^{-1} G_y \\ &= H_x P_{\chi l} C_l^T \left(C_l P_{\chi l} C_l^T + R_l \right)^{-1} G_y \\ &= H_x L_l G_y \end{aligned}$$

In this sequence of identities, the first line is a direct consequence of the foregoing lemma. To obtain the second line, we use (5.23), (5.8) and (5.7). Finally, in the third line, we have used the definition of the estimation operator L_l in (5.3).

Now we verify (5.29). We have the following sequence of identities:

$$\begin{aligned} P_{\tilde{\chi}} &= P_\chi - L C P_\chi \\ &= H_x P_{\chi l} H_x^T - H_x L_l G_y C P_\chi \\ &= H_x P_{\chi l} H_x^T - H_x L_l C_l G_x P_\chi \\ &= H_x \left(P_{\chi l} - L_l C_l P_{\chi l} \right) H_x^T \\ &= H_x P_{\tilde{\chi} l} H_x^T . \end{aligned}$$

The first line is a restatement of (5.4), while the second line uses (5.28) and (5.7). In the third line, we exploit (5.23), while in the fourth line we exploit (5.8) and (5.7). Finally, the last line represents a restatement again of (5.4).

Bibliography

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

[2] H. Akaike. "Markovian Representation of Stochastic Processes by Canonical Variables." In *SIAM Journal of Control*, vol. 13, no. 1. January, 1975.

[3] H. Akaike. "Stochastic Theory of Minimal Realization." In *IEEE Transactions on Automatic Control*, vol. 19, no. 6, December, 1974.

[4] M.R. Allen, L.E. Hoff, "Wide-Angle Wideband SAR Matched-Filter Image Formation for Enhanced Detection Performance." In *Proceedings of the SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery*, Orlando, FL, April 6-7, 1994.

[5] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Prentice-Hall, Inc., New Jersey. 1979.

[6] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis.* John Wiley & Sons, Inc., New York. 1958.

[7] K.S. Arun, B. Rao, S.Y. Kung. "A New Predictive Efficiency Criterion for Approximate Stochastic Realization." In *1983 Conference on Decision and Control*, pp. 1353-1355. 1983.

[8] K.S. Arun, S.Y. Kung. "Balanced Approximation of Stochastic Systems." In *SIAM Journal of Matrix Analysis Applications*, vol. 11, no. 1, pp. 42-68. January, 1990.

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

[10] R.D. Chaney, A.S. Willsky, L.M. Novak, "Coherent Aspect-Dependent SAR Image Formation." In *Proceedings of the SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery*, Orlando, FL, April 6-7, 1994.

[11] R. Chellappa and R. Kashyap. "Digital Image Resoration Using Spatial Interaction Models." In *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, pp. 461-472. June, 1982.

[12] R. Chellappa and S. Chatterjee. "Classification of textures using Gaussian Markov random fields." In *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, pp. 959-963, 1985.

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

[14] K.C. Chou, A.S. Willsky, and A. Benveniste. "Multiscale Recursive estimation, Data fusion and Regularization." In *IEEE Transactions on Automatic Control*, Vol. 39, No. 3, March, 1994.

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

[16] J.C. Curlander and R.N. McDonough. *Synthetic Aperture Radar, Systems and Signal Processing.* John Wiley & Sons: New York, 1991.

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

[18] H. Derin and P.A. Kelly, "Discrete-Index Markov-type Random Processes." In *Proceedings of the IEEE*, Vol. 77, No. 10, October, 1989.

[19] U.B. Desai and D. Pal. "A Realization Approach to Stochastic Model Reduction and Balanced Stochastic Realizations." In *Proceedings of the 21st Conference on Decision and Control*, pp. 1105-1114, 1982.

[20] U.B. Desai, D. Pal. "A Realization Approach to Model Reduction and Balanced Stochastic Realizations." In *IEEE 1982 Conference on Decision and Control*, pp. 1105-1122. 1982.

[21] D.E. Dudgeon and R.T. Lacoss (editors). *Lincoln Laboratory Journal–Special Issue on Automatic Target Recognition* Vol. 6, Number 1. Spring, 1993.

[22] P.W. Fieguth, W.C. Karl, A.S. Willsky and C. Wunsch, "Multiresolution Optimal Interpoltation and Statistical Analysis of TOPEX/POSEIDON Satellite Altimetry." In *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 33, No. 2, March, 1995.

[23] P.W. Fieguth, *Application of Multiscale Estimation to Large Scale Multidimensional Imaging and Remote Sensing Problems.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, June, 1995.

[24] P. Fieguth, A. Willsky, W. Karl, "Efficient Multiresolution Counterparts to Variational Methods for Surface Reconstruction," in preparation.

[25] K. Fukunaga, R.R. Hayes, L.M. Novak, "The Acquisition Probability for a Minimum Distance One-Class Classifier." In *IEEE Transactions on Aerospace and Electronic Systems.* Vol. 23, No. 4, July, 1987.

[26] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York. 1972.

[27] I.M. Gelfand and A.M. Yaglom. "Calculation of the Amount of Information about a Random Function Contained in Another Such Function." In *American Mathematical Society Transl.*, Vol. 2, No. 12, 1959.

[28] G.B. Goldstein. "False alarm regulation in log normal and Weibull clutter." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 16, January, 1973.

[29] Å. Björck and G. Golub. "Numerical Methods for Computing Angles Between Linear Subspaces." In *Mathematics of Computation.* Vol. 27, No. 123. July, 1973.

[30] S.D. Halversen, "Calculating the Orientation of a Rectangular Target in SAR Imagery." In *Proceedings of the IEEE 1992 National Aerospace and Electronics Conference* (NAECON '92), Dayton, Ohio, May 18-22, 1992.

[31] J.C. Henry, "The Lincoln Laboratory 35 GHz Airborn Polarimetric SAR Imaging System." In *IEEE National Telesystems Conference, Atlanta, GA.* page 353, March 26-27, 1991.

[32] R.A. Horn, C.R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge. 1985.

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

[34] W.W. Irving, W.C. Karl and A.S. Willsky. "A Theory for Multiscale Stochastic Realization." In *33rd Conference on Decision and Control*, December 14-16, 1994, Lake Buena Vista, FL.

[35] W.W. Irving, L.M. Novak, and A.S. Willsky. "A Multiresolution Approach to Discriminating Targets from Clutter in SAR Imagery." Submitted to *IEEE Transactions on Aerospace and Electronic Systems.*

[36] J.K. Jao, "Amplitude Distribution of Composite Terrain Radar Clutter and the K-Distribution." In *IEEE Transactions on Antennas and Propagation*, Vol. AP-32, No. 10, October, 1984.

[37] F. Jeng and J. Woods. "On the relationship of the Markov mesh to the NSHP Markov chain." *Pattern Recognition Letters*, Vol. 5, pp. 273-279, July, 1986.

[38] T. Kailath, "The Divergence and Bhattacharrya Distance Measures in Signal Selection." In *IEEE Transactions on Communication Technology*, vol. Comm-15, no. 1. February, 1967.

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

[40] D.E. Kreithen, S.D. Halversen, and G.J. Owirka, "Discriminating Targets from Clutter." In *Lincoln Laboratory Journal–Special Issue on Automatic Target Recognition*, Vol. 6, Number 1, Spring, 1993.

[41] A. Lindquist and G. Picci. "On the Stochastic Realization Problem." In *SIAM Journal of Control and Optimization*, Vol. 17, No. 3, May, 1979.

[42] L. Ljung, *System Identification: Theory for the User.* Prentice-Hall, 1987.

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

[44] M. Luettgen and A.S. Willsky, "Likelihood calculations for a class of multiscale stochastic models, with applications to texture discrimination." In *IEEE Transactions on Image Processing*, Vol. 4, No. 2, February, 1995.

[45] M. Luettgen, W.C. Karl, A.S. Willsky, and R.R. Tenney, "Multiscale representations of Markov random fields." In *IEEE Transactions on Signal Processing*, December, 1993.

[46] M. Luettgen, W. Karl, A. Willsky, "Efficient Multiscale Regularization with Applications to the Computation of Optical Flow." In *IEEE Transactions on Image Processing*, vol 3, No. 1, pp. 41-64, 1994.

[47] B. Mandelbrot and H.V. Ness, "Fractional Brownian motions, fractional noises and applications." In *SIAM Review*, vol. 10, pp. 422-436, October, 1968.

[48] M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle. "On- and off-line identification of linear state-space models." In *International Journal of Control*, vol. 49, no. 1, pp. 219-232. 1989.

[49] D.F. Morrison. *Multivariate Statistical Methods.* McGraw-Hill Book Company, New York. 1967.

[50] R.J. Muirhead. *Aspects of Multivariate Statistical Theory.* John Wiley & Sons, Inc., New York. 1982.

[51] D.C. Munson, J.D. O'Brien, and W.K. Jenkins, "A Tomographic Formulation of Spotlight Synthetic Aperture Radar." In *Proceedings of the IEEE.* Vol. 71, pp. 917-925, August, 1983.

[52] L.M. Novak, G.J. Owirka, and C.M. Netishen, "Performance of a High-Resolution Polarimetric SAR Automatic Target Recognition System." In *Lincoln Laboratory Journal–Special Issue on Automatic Target Recognition*, Vol. 6, Number 1. Spring, 1993.

[53] L.M. Novak and M.C. Burl, "Optimal Speckle Reduction in Polarimetric SAR Imagery." In *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 2, March, 1990.

[54] P. Van Overschee, B. De Moor. "Subspace Algorithms for the Stochastic Identification Problem." In *Automatica*, vol. 29, no. 3, pp. 649-660. 1993.

[55] A.P. Pentland "Fractal-based description of natural scenes." In *IEEE Transactions of Pattern Analysis and Machine Intelligence.* Vol. 6, pp.661-674. November, 1984.

[56] G. Picci. *Stochastic Realization of Gaussian Processes.* In *Proceedings of the IEEE*, Vol. 64, No. 1, January, 1974.

[57] W.H. Press, S.A. Teukolsky, W T. Vetterling and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, Second Edition, 1992.

[58] H.E. Rauch, F. Tung and C.T. Striebel. "Maximum Likelihood Estimates of Linear Dynamic Systems." In *AIAA Journal*, Vol. 3, No. 8, August, 1965.

[59] M.A. Sironvalle. "The Random Coin Method: Solution of the Problem of Simulation of a Random Function in the Plane." In *Mathematical Geology*, vol. 12, no. 1, 1980.

[60] G. Strang, *Linear Algebra and its Applications*. Harcourt Brace Jovanovich, Publishers, San Diego, 1988.

[61] N.S. Subotic, L.M. Collins, J.D. Gorman, and B.J. Thelen, "A Multiresolution Approach to Target Detection in Synthetic Aperture Radar Data." In *1994 Asilomar Conference Proceedings*, Monterey, CA, October, 1994.

[62] A. VanDerVeen, E.F. Deprettere, A.L. Swindlehurst, "Subspace-Based Signal Analysis Using Singular Value Decomposition." In *Proceedings of the IEEE*. Vol. 81, No. 9. September, 1993.

[63] H.L. VanTrees. *Detection, Estimation and Modulation Theory: Part I.* Wiley, New York, NY, 1968.

[64] A.L. van den Wollenberg. "Redundancy Analysis: An Alternative for Canonical Correlation Analysis." In *Psychometrika*, vol. 42, no. 2, June, 1977.

[65] J.W. Woods. "Two-dimensional discrete Markovian fields." In *IEEE Transactions on Information Theory*, Vol. 18, March, 1972.

[66] J.W. Woods and C.H. Radewan. "Kalman Filtering in Two Dimensions." In *IEEE Transactions on Information Theory*, Vol. 23, No. 4, July, 1977.

[67] A.M. Yaglom. *Correlation Theory of Stationary and Related Random Functions I.* Springer-Verlag, New York, 1987.

[68] S.H. Yueh, J.A. Kong, J.K. Jao, R.T. Shin and L.M. Novak, "K-distribution and polarimetric terrain radar clutter." In *Journal of Electromagnetic Waves and Applications*, Vol. 3, No. 8, 1989.