LIDS-P-2085

MULTISCALE RECURSIVE ESTIMATION, DATA FUSION, AND REGULARIZATION

Kenneth C. Chou[1]
Alan S. Willsky[1]
Albert Benveniste[2]

Abstract

A current topic of great interest is the multiresolution analysis of signals and the development of multiscale signal processing algorithms. In this paper we describe a framework for modeling stochastic phenomena at multiple scales and for their efficient estimation or reconstruction given partial and/or noisy measurements which may also be at several scales. In particular multiscale signal representations lead naturally to pyramidal or tree-like data structures in which each level in the tree corresponds to a particular scale of representaion. Noting that scale plays the role of a time-like variable, we introduce a class of multiscale dynamic models evolving on dyadic trees. The main focus of this paper is on the description, analysis, and application of an extremely efficient optimal estimation algorithm for this class of models. This algorithm consists of a fine–to–coarse filtering sweep, followed by a coarse–to–fine smoothing step, corresponding to the dyadic tree generalization of Kalman filtering and Rauch–Tung–Stiebel smoothing. The Kalman filtering sweep consists of the recursive application of 3 steps: a measurement update step, a fine–to–coarse prediction step, and a fusion step, the latter of which has no counterpart for time- (rather than scale-) recursive Kalman filtering. We illustrate the use of our methodology for the fusion of multiresolution data and for the efficient solution of "fractal regularizations" of ill–posed signal and image processing problems encountered, for example, in low–level computer vision.

[1] Laboratory for Information and Decision Systems and Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge,MA 02139, USA. The work of these authors was also supported in part by the Air Force Office of Scientific Research under Grant AFOSR-88-0032, by the National Science Foundation under Grants MIP-9015281 and INT-9002393, and by the Office of Naval Research under Grant N00014-91-J-1004. In addition some of this research was performed while KCC and ASW were visitors at IRISA and while ASW received support from INRIA. K.C. Chou is presently with SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025.

[2] Institut de Recherche en Informatique et Systemes Aleatoires (IRISA), Campus de Beaulieu, 35042 Rennes, CEDEX, FRANCE.The research of this author was also supported in part by Grant CNRS G0134.

1 Introduction

A topic of considerable current interest and activity in the signal and image processing communities is the development of multiresolution processing algorithms. Much of the motivation for this has come directly from the fact that many phenomena exhibit distinctive patterns at multiple scales of resolution. For example fractal models, such as stochastic processes with 1/f–like spectral characteristics, have often been suggested for the description of natural scenes and images [2,22,23,25]. Also, transient events and spatially–localized features can naturally be thought of as the superposition of fine resolution detail on a more coarsely varying background. A second motivation for this research is purely computational: many signal analysis problems, especially those in several spatial dimensions, are of enormous computational complexity, and thus extremely efficient, possibly highly parallel algorithms are essential. Given the success of multigrid methods for solving partial differential equations [8,9,31], the use of pyramidal, multiscale signal representations would seem to offer considerable promise as a framework for efficient algorithm development.

Multiresolution signal and image representation and analysis methods have been investigated for some time under a variety of names including multirate filters [41,35], subband coding [28], Laplacian pyramids [10], and "scale–space" image processing [38]. However it is the emerging theory of wavelet transforms [14,15,18,20,21,24,29] that has sparked much of the recent flurry of activity in this area, in part because of its rich mathematical foundation and in part because of the evocative examples and research on wavelets indicating that it should be possible to develop efficient, optimal processing algorithms based on such multiscale representations. The development of such optimal algorithms- e.g., for the reconstruction of noise degraded signals or for the localization of transient signals- and the evaluation of their performance requires the development of a corresponding multiscale theory of stochastic processes that allows us both to model phenomena and to develop efficient algorithms for their estimation and analysis. The research presented in this paper and in several others [3,4,5,6,11] has the development of such a theory as its objective.

In particular in this paper we introduce a class of multiscale state space models. The standard time domain versions of these models have, of course, proven to be of considerable value both because of the extremely efficient algorithms they admit (such as the Kalman filter) and because rich classes of stochastic phenomena can be well-modeled using such state space descriptions. As we will see both of these are also true for our multiscale state models, leading to the possibility of devising novel and extremely efficient algorithms for a variety of signal and image analysis problems. The key to our development is the observation that multiscale signal representations, whether for 1-D time series or multidimensional images, have a natural, time-like variable associated with them, namely scale. In particular essentially all methods for representing and processing signals at multiple scales involve pyramidal data structures, where each level in the pyramid corresponds to a particular scale and each node at a given scale is connected both to a parent node at the next coarser scale and to several descendent nodes at the next finer scale.

In 1-D, if the typical scale–to–scale decimation by a factor of two is used, we are led directly to a dyadic tree data structure. The simplest example of such a data structure is provided by the Haar wavelet representation in which the representation of a signal $f(x)$ at the mth scale is given by its average values $f(m,n)$ over successive intervals of length 2^{-m}, i.e.

$$f(m,n) = k_m \int_{n2^{-m}}^{(n+1)2^{-m}} f(x)dx \tag{1.1}$$

where k_m is a normalizing constant. In this case each node (m,n) is connected to a single "parent" node $(m-1,[\frac{n}{2}])$, where $[y]$ = the integer part of y, and to two "descendent" nodes $(m+1,2n)$, $(m+1,2n+1)$, where the fine–to–coarse relationship among the $f(m,n)$ values is given by an interpolation plus the adding of higher resolution detail not available at the coarser level.

There are several important points to note from this simple example. The first is that the model relating the signal representations at different scales consists of a local scale–to–scale recursion on the dyadic tree structure of the (m, n) index set. Secondly, while the fine–to–coarse recursion corresponds to the multiresolution analysis of signals, the coarse–to–fine recursion, in which we add higher resolution detail at each scale, corresponds to the multiresolution synthesis of signals. It is the latter of these, then, that is appropriate for the multiresolution modeling of signals and phenomena. In doing this, however, we wish to consider a broader class of multiscale representations than those provided by the Haar transform. In particular we choose to view each scale of such a representation more abstractly, much as in the notion of state, as capturing the feature of signals up to that scale that are relevant for the "prediction" of finer scale approximations. For example the Haar transform recursion can naturally be thought of as a first–order recursion in scale. However, as we know from time series analysis, a considerably broader class of analytically tractable models is obtained if we allow higher–order dynamics, i.e. by introducing additional memory in scale.

In [3,4] we investigate such an extension by developing a theory of multiscale autoregressive modeling. In this paper we develop an alternate and in fact more general modeling framework by considering vector state space recursive models on the dyadic tree. As we will see, by doing this we provide a framework that allows us to model phenomena with multiscale features and to develop extremely efficient, parallelizable algorithms. Moreover, by adopting this perspective of multiscale modeling we provide a natural setting not only for dealing with multiscale phenomena and multiscale algorithms but also with multiscale data. In particular a problem of considerable importance in a variety of applications (e.g. remote sensing) is the fusion of data from heterogeneous suites of sensors (multicolor IR, visual, microwave, etc.) which in general provide information in different spectral bands and at different resolutions. The framework we describe offers the possibility of modeling such multiresolution data simply as measurements at different levels in the dyadic tree, resulting in data fusion algorithms that are no more complex then algorithms for filtering single resolution data. Furthermore since the key to our models and algorithms are recursions in scale, we obtain essentially the same algorithmic structures for two- or higher-dimensional image processing. For example in 2-D our dyadic tree would be replaced by a quadtree in which each node has 4 descendants rather than 2, resulting in an increase in computational complexity of a factor of 2 in going from 1-D to 2-D rather than the usual geometric growth encountered in other multidimensional signal processing formalisms. The use of a simple quadtree model of the type described in this paper has been explored in the context of image coding and reconstruction in [12,33]. The work described here provides a general formalism and machinery for such models which appears to hold promise for a variety of applications.

In the next section we introduce our multiresolution state space models and perform some elementary statistical analysis. In Section 3 we then investigate the problem of optimal estimation for these multiscale models and in particular develop the generalization of the Rauch–Tung–Striebel (RTS) algorithm [26], consisting of a fine–to–coarse filtering sweep followed by a coarse–to–fine smoothing sweep. The fine–to–coarse sweep, corresponding to a generalization of the Kalman filter to multiscale models on trees, consists of a 3-step recursion of measurement updating, fine–to–coarse prediction, and the fusion of information as we move from fine–to–coarse scales. The last of these 3 steps has no counterpart in standard Kalman filtering, and this in turn leads to a new class of scale-recursive Riccati equations for the filtering error covariance. In Section 4 we then illustrate the application of our methodology for the estimation of both fractal, 1/f–like processes and standard Gauss-Markov processes, both based on single-scale measurements and based on the fusion of multiresolution data. In addition we demonstrate the use of our methodology for the efficient solution of a "fractal regularization" of a motion estimation problem typical of many ill-posed image processing problems encountered, for example, in low-level computer vision. For simplicity our entire development is carried on in the context of the dyadic tree which corresponds to the representation and processing of 1-D signals. As we have indicated the extension to higher dimensions introduces only notational rather than analytical or comptutational complexity. This, together with the other examples in this section point to the potential for this formalism both to provide solutions to a variety of challenging estimation and data fusion problems and to yield substantial savings for many image processing problems. In addition, as we have indicated, the estimation algorithm developed in this paper leads to a new class of Riccati equations. In [43] we develop several basic system-theoretic concepts for systems on trees and use these to analyze these Riccati equations and the error dynamics of the filtering algorithm on trees developed here.

2 State Space Models on Dyadic Trees

As discussed in the preceding section, the focus of attention in this paper is on stochastic process and dynamic models defined on the dyadic tree of scale/translation pairs (m, n) where each value of m corresponds to a particular scale of resolution and there is a factor of two decimation between any scale and its neighboring scale. The structure of this tree is illustrated in Fig. 1 where for notational convenience we denote each node of the tree by a single abstract index t, i.e. $t = (m, n)$, where T denotes the set of all nodes, and $m(t)$ denotes the scale or m-component of t. In order to define dynamics on T we need to introduce the basic shift operators on T, which play roles analogous to forward and backward shifts for temporal systems. In particular, with increasing m— i.e. the coarse–to–fine direction of synthesis— denoting the forward direction, we can define a unique backward shift $\overline{\gamma}$ and two forward shifts α and β (see Figure 1). In particular if $t = (m, n)$, then $t\alpha = (m+1, 2n)$, $t\beta = (m+1, 2n+1)$, and $t\overline{\gamma} = (m-1, [\frac{n}{2}])$. Note that $\overline{\gamma}$ is onto but not one–to–one, as one would expect of what is basically a fine–to–coarse decimination step, while α and β, corresponding to coarse–to–fine interpolation steps, are one–to–one but not onto.

The structure of T naturally admits two alternate classes of scale-recursive linear dynamic models that are defined locally on T. The first of these is the class of coarse–to–fine state space models on T :

$$x(t) = A(t)x(t\overline{\gamma}) + B(t)w(t) \tag{2.1}$$

$$y(t) = C(t)x(t) + v(t) \tag{2.2}$$

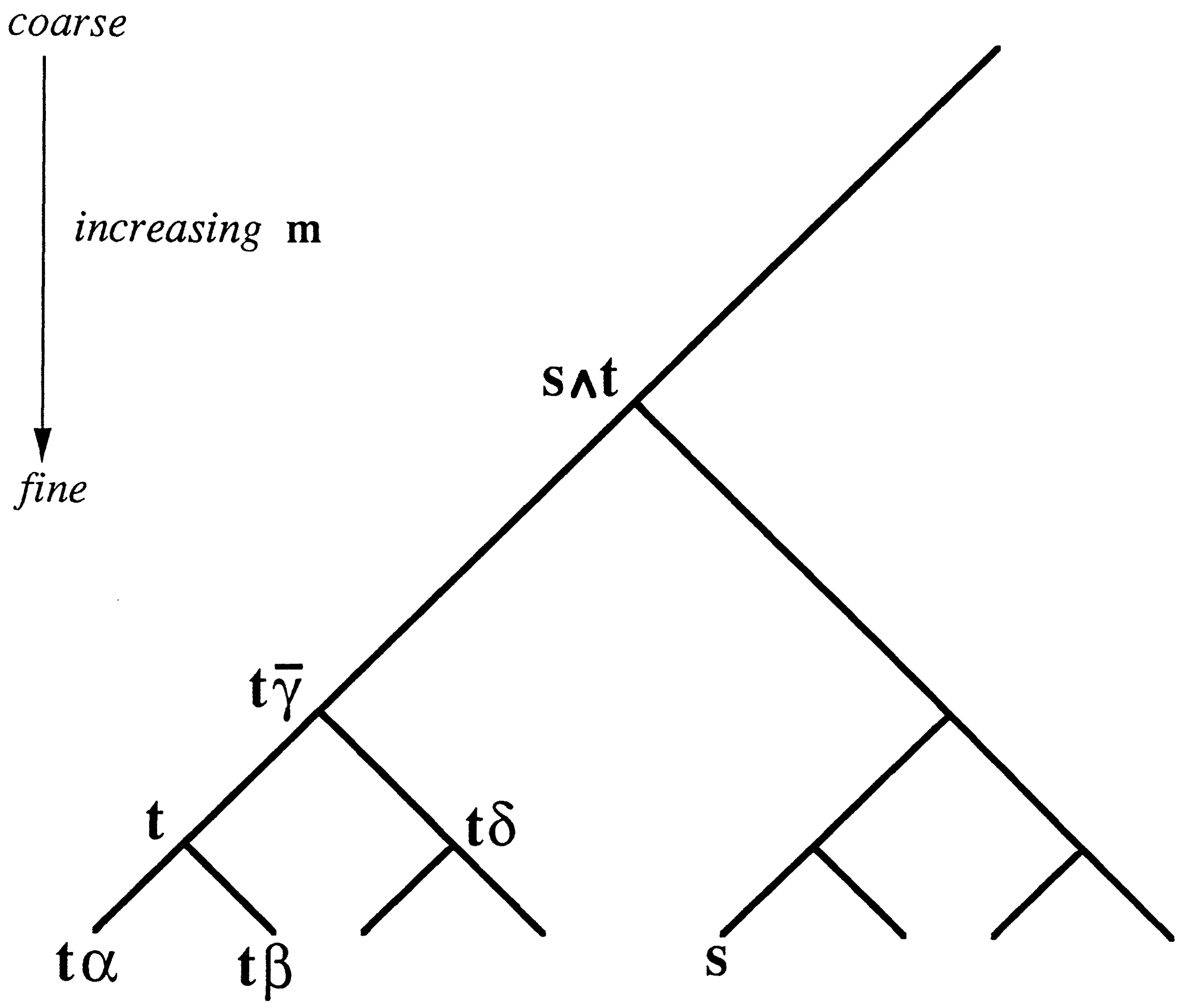


Figure 1: Illustrating the multiscale structure of the dyadic tree and some notation used in the paper.

where $w(t)$ and $v(t)$ are independent, zero-mean vector white noise processes with covariances I and $R(t)$, respectively, and $x(t)$ is an n-dimensional, zero-mean stochastic process. The term $A(t)x(t\bar{\gamma})$ in (2.1) represents a coarse–to–fine prediction or interpolation, $B(t)w(t)$ represents the higher resolution detail added in going from one scale to the next finer scale, and $y(t)$ is the measured variable (if any) at the particular scale m and location n represented by t. Thus (2.1) represents a natural generalization of coarse–to–fine synthesis form of the wavelet transform in that we allow additional memory, captured by the vector $x(t)$, at each scale, together with a general scale–to–scale linear recursion rather than the particular synthesis recursion arising in the Haar transform. This form serves as the basis for our multiscale modeling of stochastic processes and also will arise in the coarse–to–fine smoothing sweep of the RTS algorithm developed in Section 3. In contrast the fine–to–coarse Kalman filtering step of our RTS algorithm falls into the class of fine–to–coarse recursive models of the form

$$x(t) = F_1(t\alpha)x(t\alpha) + F_2(t\beta)x(t\beta) + G(t\alpha)w(t\alpha) + G(t\beta)w(t\beta) \tag{2.3}$$

which represents a vector generalization of the fine–to–coarse averaging and decimation operation of the analysis form of the Haar transform.

Note that the general models (2.1)-(2.3) allow full t-dependence of all the system matrices - e.g., $A(t)$ can vary with both scale and translational position. An important special case is that in which the system parameters are constant at each scale but may vary from scale to scale, in which case we abuse notation by writing $A(t) = A(m(t))$, etc. Such a model is useful for capturing the fact that data may be available at only particular scales (i.e. $C(m) \neq 0$ only for particular values of m); for example in the original context of wavelet analysis, we actually have only one measurement set, corresponding to $C(m)$ being nonzero only at the finest scale in our representation. Also, by varying $A(m), B(m)$, and $R(m)$ with m we can capture a variety of scale-dependent effects. For example, dominant scales might correspond to scales with larger values of $B(m)$. Also, by building a geometric decay in scale into $B(m)$ it is possible to capture 1/f-like, fractal behavior as shown and studied in

[36,37].

The general case of t-varying parameters also has a number of potential uses. For example such a form for $C(t)$ is clearly required in data fusion problems in which coarse-resolution, broad coverage data (such as satellite IR images of ocean temperature) are to be fused with sparse but high resolution data (such as from ship-borne temperature measurements). An example illustrating this is given in Section 4 where we also encounter the need for full t-dependence in the context of the low-level image processing problem of motion estimation. Thus in Section 3 we describe the RTS algorithm for the general t-varying case.

In the remainder of this section we analyze the basic statistics of the model (2.1). For reasons that will become clear we assume that $A(t)$ is invertible for all t. The analysis and results we present extend to the case when this is not true, but our discussion is simplified if we make this assumption. One additional assumption we make is that $w(t)$ is independent of the "past" of x, i.e. $\{x(\tau)|m(\tau) < m(t)\}$. Thanks to the invertibility of $A(t)$, this is equivalent to requiring $w(t)$ to be independent of *some* $x(\tau)$ with $\tau \neq t, m(\tau) < m(t)$. In this case (2.1) not only describes a scale–to–scale Markov process, but it in fact specifies a Markov random field on T in that conditioned on $x(t\overline{\gamma}), x(t\alpha)$, and $x(t\beta)$, $x(t)$ is independent of x at all other nodes.[1]

Finally, let us comment on the domain over which (2.1) is defined. In particular if we wish to consider representations of signals of unbounded extent, we must deal with the full infinite tree T, i.e. $\{(m,n)| -\infty < m,n < \infty\}$. This is of interest only when we consider asymptotic properties such as stability and steady-state behavior. In any practical application, of course, we must deal with a compact interval of data. In this case the index set of interest represents a finite version of the tree of Figure 1, consisting of $M+1$ levels beginning with the coarsest scale represented by a unique root node, denoted by 0, and M subsequent levels, the finest of which has 2^M nodes. In this case, we simply assume that $w(t)$ is independent of the initial condition $x(0)$ which is asumed to have zero mean and covariance $P_x(0)$.

[1] Indeed this fact is used in [42] to describe a multigrid-like iterative algorithm for the solution of the multiscale estimation problem studied in this paper.

Straightforward calculations, analoguous to those for standard state space models, yield the second-order statistics of $x(t)$. In particular the covariance $P_x(t) = E[x(t)x^T(t)]$ evolves according to a Lyapunov equation on the tree:

$$P_x(t) = A(t)P_x(t\bar{\gamma})A^T(t) + B(t)B^T(t) \tag{2.4}$$

Let $K_{xx}(t,s) = E[x(t)x^T(s)]$. Let $s \wedge t$ denote the least upper bound of s and t, i.e. the first node that is a predecessor of both t and s. Then

$$K_{xx}(t,s) = \Phi(t, s \wedge t)P_x(s \wedge t)\Phi^T(s, s \wedge t) \tag{2.5}$$

where for $m(t_1) \geq m(t_2)$

$$\Phi(t_1,t_2) = \begin{cases} I & t_1 = t_2 \\ A(t_1)\Phi(t_1\bar{\gamma}, t_2) & m(t_1) > m(t_2) \end{cases} \tag{2.6}$$

Note that (2.5) differs from the formula for standard state models in which the state transition matrix appears on either the left or right (depending on whether $t < s$ or $s < t$), but not both sides of P_x.

In the scale-varying model, *i.e.* the case in which the model parameters vary in scale only, if at some scale $P_x(t)$ is constant, then this holds at each scale, so that by an abuse of notation $P_x(t) = P_x(m(t))$, and we have a scale–to–scale Lyapunov equation:

$$P_x(m+1) = A(m)P_x(m)A^T(m) + B(m)B^T(m) \tag{2.7}$$

(Note that this is always true if we are considering a finite subtree with single root node 0). Our covariance, $K_{xx}(t,s)$, in this case has the following form.

$$K_{xx}(t,s) = \Phi(m(t), m(s \wedge t))P_x(m(s \wedge t))\Phi^T(m(s), m(s \wedge t)) \tag{2.8}$$

where for $m_1 \geq m_2$

$$\Phi(m_1,m_2) = \begin{cases} I & m_1 = m_2 \\ A(m_1)\Phi(m_1 - 1, m_2) & m_1 > m_2 \end{cases} \tag{2.9}$$

Figure 2(a) depicts the sample paths of a scalar example of such a model with $A = .9$ and $B(m) = 2^{-m/2}, m = 1, ..., 7$, and $P_x(0) = 1$. As we have indicated, the use of a geometrically–varying noise gain allows us to capture fractal signal characteristics with commensurately–scaled fluctuations at all scales.

If we further specialize our model to the case in which A and B are constant, we encounter an interesting notion of stationarity. Specifically if A is stable, then (2.7) admits a steady–state solution, to which $P_x(m)$ converges, which is the unique solution of the usual algebraic Lyapunov equation:

$$P_x = AP_xA^T + BB^T \tag{2.10}$$

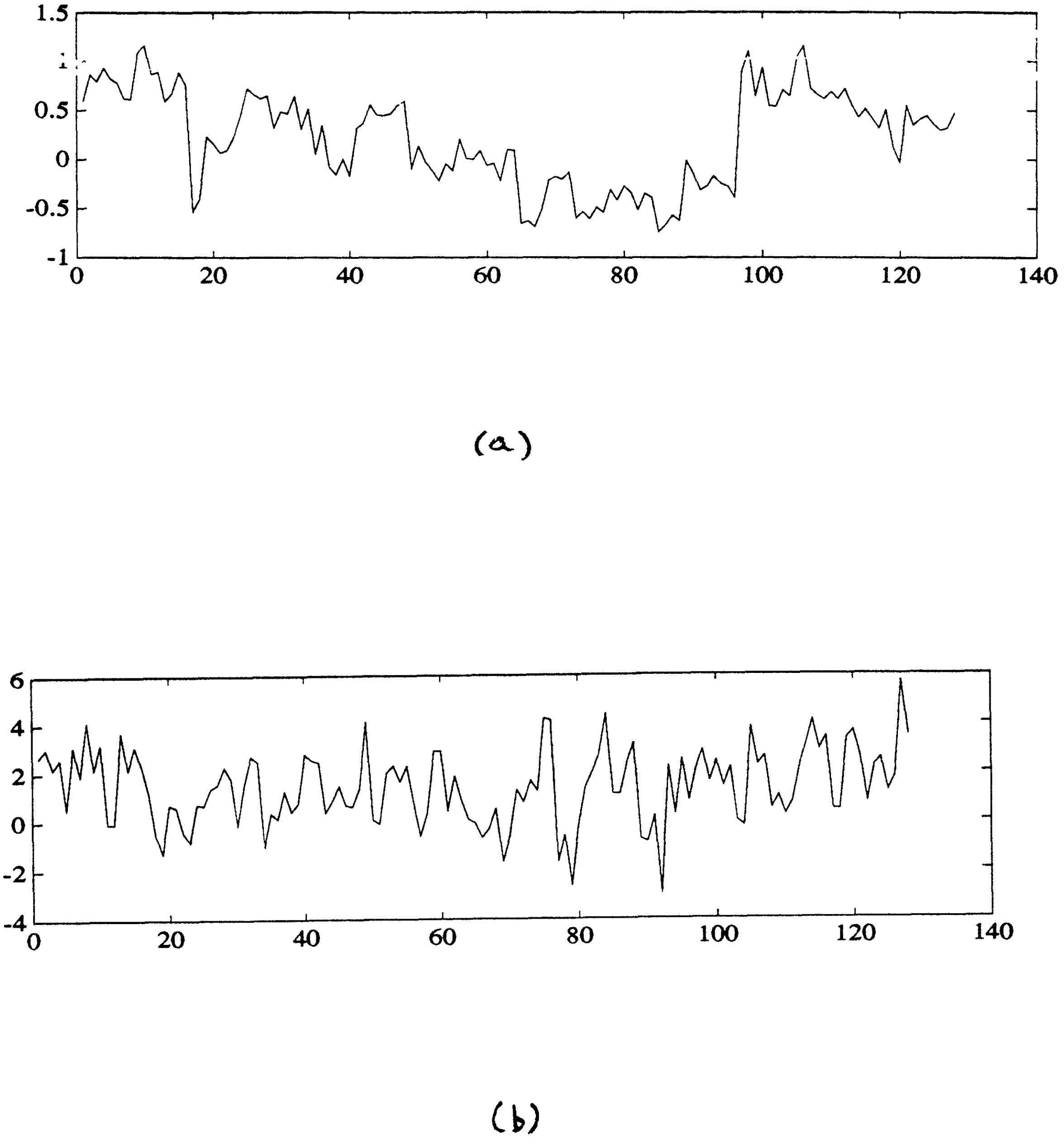
In this case if $P_x(0) = P_x$(if we have a root node), or if we assume that $P_x(\tau) = P_x$ for $m(\tau)$ sufficiently negative, then $P_x(t) = P_x$ for all t, and we have what we refer to as a stationary model

$$\begin{aligned} K_{xx}(t,s) &= A^{d(t,s\wedge t)} P_x (A^T)^{d(s,s\wedge t)} \\ &= K_{xx}(d(t, s\wedge t), d(s, s\wedge t)) \end{aligned} \tag{2.11}$$

where $d(t_1, t_2)$ denotes the distance between t_1 and t_2, *i.e.* the number of branches on the shortest path from t_1 to t_2. Figure 2(b) depicts the sample path of a scalar example of such a model with $A = .9, B = P_x(0) = 1$.

Referring to Figure 1 we see that the covariance in (2.11) only depends upon the distances of s and t to their commmon "parent" node $s \wedge t$, *i.e.* the finest scale node that has both s and t as descendents. This yields a notion of self–similarity since, roughly speaking (2.11) states that the correlation between $x(t)$ and $x(s)$ depends on the differences in scale and in temporal offset of the nodes t and s. A stronger notion of shift invariance for stochastic processes on trees is characterized by the condition that $K_{xx}(t,s)$ depends only on $d(s,t)$, and we refer to such processes as *isotropic*. Note that (2.11) represents an isotropic covariance if $AP_x = P_xA^T$, which points to the connection to the class of reversible stochastic processes[1]. For example in the scalar case

$$K_{xx}(t,s) = \left\{ \frac{B^2}{1-A^2} \right\} A^{d(s,t)} \tag{2.12}$$



(a)

(b)

Figure 2: Sample paths for two scalar multiscale models:
(a) Process with A= .9, B(m) = $2^{-m/2}$, $P_x(0)$ = 1;
(b) Process with A= .9, B=$P_x(0)$ = 1

Note that since $d(s,t) = d(s, s \wedge t) + d(t, s \wedge t)$, any isotropic process is stationary, but the reverse implication is not always true. We refer the reader to [5,6] for a detailed analysis of isotropic processes and in particular for the development of a theory for their multiscale autoregressive modeling.

Finally, in our development of smoothing algorithms in the next section we will encounter the need for fine–to–coarse prediction and recursion. In particular, we will need to consider the reversal of (2.1), *i.e.* a model representing $x(t\bar{\gamma})$ as a linear function of $x(t)$ and a noise that is uncorrelated with $x(t)$. To do this, we can directly apply the results of [34]:

$$x(t\bar{\gamma}) = F(t)x(t) - A^{-1}(t)B(t)\tilde{w}(t) \tag{2.13}$$

with

$$\begin{aligned} F(t) &= A^{-1}(t)[I - B(t)B^T(t)P_x^{-1}(t)] \\ &= P_x(t\bar{\gamma})A^T(t)P_x^{-1}(t) \end{aligned} \tag{2.14}$$

and where

$$\tilde{w}(t) = w(t) - E[w(t)|x(t)] \tag{2.15}$$

$$\begin{aligned} E[\tilde{w}(t)\tilde{w}^T(t)] &= I - B^T(t)P_x^{-1}(t)B(t) \\ &\triangleq \tilde{Q}(t) \end{aligned} \tag{2.16}$$

In the development of reverse–time models for standard space models the noise process $\tilde{w}(t)$ is white in time. In our case the situation is a bit more complex. In particular $\tilde{w}(t)$ *is* white along all upward paths on the tree— *i.e.* $\tilde{w}(s)$ and $\tilde{w}(t)$ are uncorrelated if $s \wedge t = s$ or t. Otherwise it is not difficult to check that $\tilde{w}(s)$ and $\tilde{w}(t)$ are *not* uncorrelated.[2] Indeed in the stationary case, $\tilde{w}(t)$ is also stationary with correlation matrix $K(d_1, d_2)$ that *does* equal zero if d_1 or d_2 (but not both) are zero but that is *not* zero if both d_1 and d_2 are nonzero.

[2] In fact $\tilde{w}(t)$ is a martingale difference for a martingale defined on the partially-ordered tree [39].

3 A Two-Sweep Estimation Algorithm for Multiscale Processes

In this section we derive an extremely efficient algorithm for the smoothing of (possibly) multiscale measurement data for our dynamic system (2.1), (2.2) on the dyadic tree. The algorithm is a generalization of the well-known Rauch-Tung-Striebel(RTS) smoothing algorithm for causal state models. Recall that the standard RTS algorithm involves a forward Kalman filtering sweep followed by a backward sweep to compute the smoothed estimates. The generalization to our models on trees has the same structure, with several important differences. First for the standard RTS algorithm the procedure is completely symmetric with respect to time – i.e. we can start with a reverse-time Kalman filtering sweep followed by a forward smoothing sweep. For processes on trees, the Kalman filtering sweep **must** proceed from fine-to-coarse (i.e. in the reverse direction from that in which the model (2.1) is defined) followed by a coarse-to-fine smoothing sweep[3]. Furthermore the Kalman filtering sweep, using the backward model (2.13)–(2.16) is somewhat more complex for processes on trees. In particular one full step of the Kalman filter recursion involves a measurement update, **two** parallel backward predictions (corresponding to backward prediction along both of the paths descending from a node), and the **fusion** of these predicted estimates. This last step has no counterpart for state models evolving in *time* and is one of the major reasons for the differences between the analysis of temporal Riccati equations and that presented in the sequel [43] to this paper. As a final remark we note that our algorithm has a pyramidal structure consistent with that of the tree and thus has considerable parallelism. In particular such a structure maps naturally onto the hypercube multiprocessor architecture.

We begin by describing the fine–to–coarse filtering step, in which we recursively compute the optimal estimate of $x(t)$ based on data in the subtree with root node t

[3] The reason for this is not very complex. To allow the measurement on the tree at one point to contribute to the estimate at another point on the same level of the tree, one must use a recursion that first moves up and then down the tree. Reversing the order of these steps does not allow one to realize such contributions.

(see Figure 3). Specifically, let us define some notation:

$$\begin{aligned} Y_t &= \{y(s) | s = t \text{ or } s \text{ is a descendant of } t\} \\ &= \{y(s) | s \in t(\alpha, \beta)^*\} && (3.1) \\ Y_t^+ &= \{y(s) | s \in t\alpha(\alpha, \beta)^* \text{ or } t\beta(\alpha, \beta)^*\} && (3.2) \\ \hat{x}(\cdot|t) &= E[x(\cdot)|Y_t] && (3.3) \\ \hat{x}(\cdot|t+) &= E[x(\cdot)|Y_t^+] && (3.4) \end{aligned}$$

The interpretation of these estimates is provided in Figure 3. We begin by considering the measurement update step. Specifically, suppose that we have computed $\hat{x}(t|t+)$ and the corresponding error covariance, $P(t|t+)$. Then, standard estimation results yield

$$\begin{aligned} \hat{x}(t|t) &= \hat{x}(t|t+) + K(t)[y(t) - C(t)\hat{x}(t|t+)] && (3.5) \\ K(t) &= P(t|t+)C^T(t)V^{-1}(t) && (3.6) \\ V(t) &= C(t)P(t|t+)C^T(t) + R(t) && (3.7) \end{aligned}$$

and the resulting error covariance is given by

$$P(t|t) = [I - K(t)C(t)]P(t|t+) \qquad (3.8)$$

Suppose now that we have computed $\hat{x}(t\alpha|t\alpha)$ and $\hat{x}(t\beta|t\beta)$. Note that $Y_{t\alpha}$ and $Y_{t\beta}$ are disjoint and these estimates can be calculated in parallel. We then compute $\hat{x}(t|t\alpha)$ and $\hat{x}(t|t\beta)$ which are given by

$$\begin{aligned} \hat{x}(t|t\alpha) &= F(t\alpha)\hat{x}(t\alpha|t\alpha) && (3.9) \\ \hat{x}(t|t\beta) &= F(t\beta)\hat{x}(t\beta|t\beta) && (3.10) \end{aligned}$$

with corresponding error covariances given by

$$\begin{aligned} P(t|t\alpha) &= F(t\alpha)P(t\alpha|t\alpha)F^T(t\alpha) + Q(t\alpha) && (3.11) \\ Q(t\alpha) &= A^{-1}(t\alpha)B(t\alpha)\tilde{Q}(t\alpha)B^T(t\alpha)A^{-T}(t\alpha) && (3.12) \\ P(t|t\beta) &= F(t\beta)P(t\beta|t\beta)F^T(t\beta) + Q(t\beta) && (3.13) \\ Q(t\beta) &= A^{-1}(t\beta)B(t\beta)\tilde{Q}(t\beta)B^T(t\beta)A^{-T}(t\beta) && (3.14) \end{aligned}$$

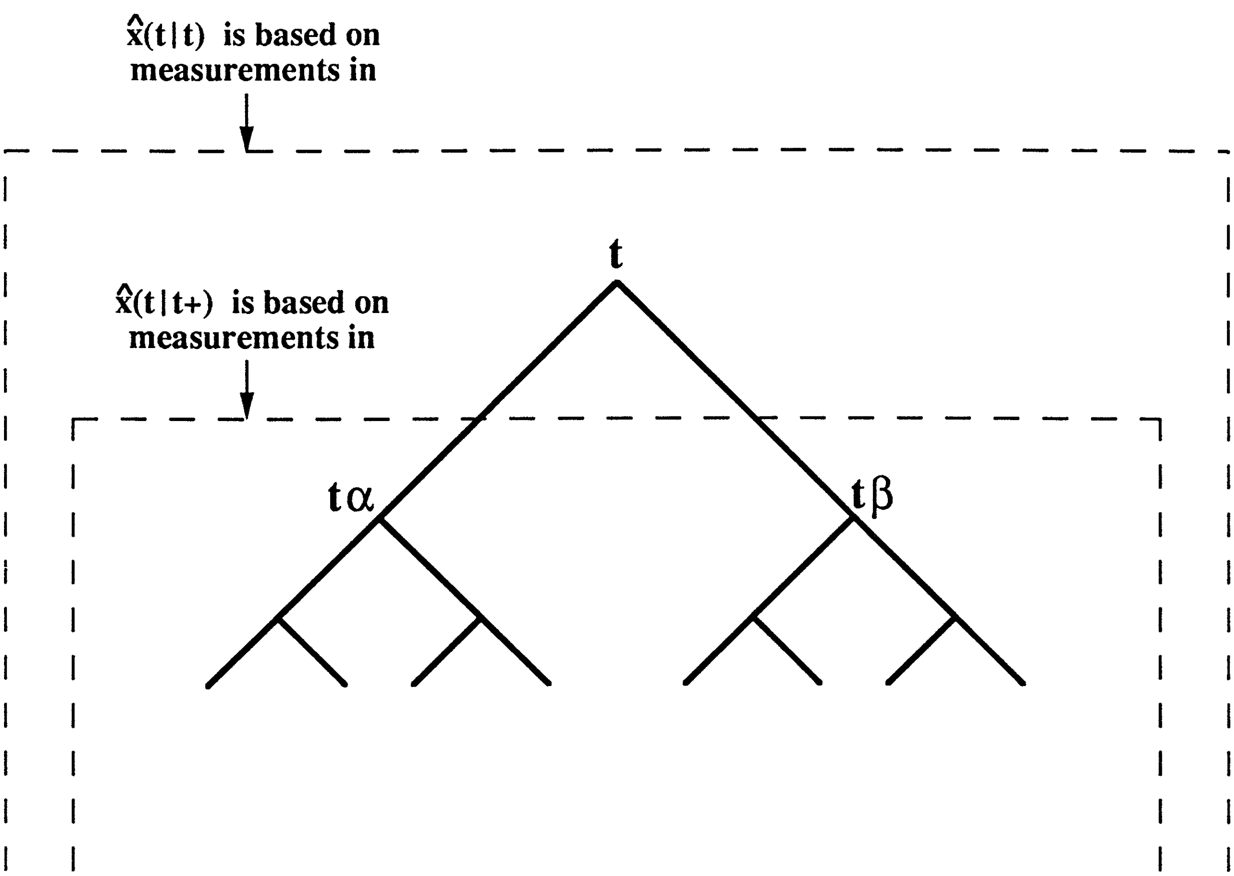

$\hat{x}(t|t)$ is based on measurements in
$\hat{x}(t|t+)$ is based on measurements in
t
tα
tβ

Figure 3: Representation of Measurement Update and Merged Estimates

These equations follow directly from the backward model (2.13)-(2.16).

The third and final step of the Kalman filtering recursion is to merge the estimaters (3.9) and (3.10), to form $\hat{x}(t|t+)$:

$$\hat{x}(t|t+) = P(t|t+)[P^{-1}(t|t\alpha)\hat{x}(t|t\alpha) + P^{-1}(t|t\beta)\hat{x}(t|t\beta)] \tag{3.15}$$

$$P(t|t+) = [P^{-1}(t|t\alpha) + P^{-1}(t|t\beta) - P_x^{-1}(t)]^{-1} \tag{3.16}$$

The interpretation of (3.15), (3.16) is rather simple: $\hat{x}(t|t\alpha)$ ($\hat{x}(t|t\beta)$) is the best estimate of $x(t)$ based on the prior statistics of $x(t)$ (i.e. it is zero–mean, with covariance $P_x(t)$) and the measurements in $Y_{t\alpha}$ (in $Y_{t\beta}$); consequently the fusion of these estimates, as captured in (3.15) and in particular (3.16), must avoid a double–counting of prior information. A brief proof of (3.15), (3.16) is given in Appendix *A*.

Equations (3.5)-(3.16) define the coarse–to–fine Kalman filter for our multiscale stochastic model. The on–line calculations consist of an update step (3.5), a pair of parallel prediction steps (3.9), (3.10), and the fusion step (3.15). The associated Riccati equation consists of three corresponding steps (3.6)-(3.8), (3.11)-(3.14), (3.16) the first two of which correspond to the usual Riccati equation (in reverse time). The third step (3.16) has no counterpart in the standard case.

Let us now turn to the smoothing problem, i.e. the computation of $\hat{x}_s(t)$, the optimal estimate of $x(t)$ based on all available data. As in the usual RTS algorithm, this requires focusing on a compact data set, here corresponding to a finite subtree with root node 0 and M scales below it. The initialization of the Kalman filter in this case at scale $m(t) = M$ is given by $\hat{x}(t|t+) = 0, P(t|t+) = P_x(t)$. Once the Kalman filter has reached the root node at the top of the tree, we have computed $\hat{x}_s(0) = \hat{x}(0|0)$, which serves as the initial condition for the coarse–to–fine smoothing sweep which also has a parallel, pyramidal structure. Specifically, suppose that we have computed $\hat{x}_s(t\bar{\gamma})$. This is then combined with the fine–to–coarse filtered estimate $\hat{x}(t|t)$ to produce $\hat{x}_s(t)$:

$$\hat{x}_s(t) = \hat{x}(t|t) + J(t)\,[\hat{x}_s(t\bar{\gamma}) - \hat{x}(t\bar{\gamma}|t)] \tag{3.17}$$

where

$$J(t) \;\triangleq\; P(t|t)F^T(t)P^{-1}(t\bar{\gamma}|t) \tag{3.18}$$

We also have a coarse–to–fine recursion for the smoothing error covariance, initialized with $P_s(0) = P(0|0)$:

$$P_s(t) = P(t|t) + J(t)[P_s(t\bar{\gamma}) - P(t\bar{\gamma}|t)]J^T(t) \tag{3.19}$$

Equations (3.17)-(3.19) are of exactly the same form as the usual RTS smoothing sweep, although the derivation, which is described in Appendix B, is somewhat more involved.

4 Examples and Applications

In this section we provide several illustrations of the application of the estimation framework described in this paper. The purpose of the first subsection is to demonstrate that the highly parallelizable estimation structure we have developed can be succesfully applied to a rather rich class of processes extending well beyond those that are precisely of the form generated by our tree models. In the second subsection we present examples of the fusion of multiresolution data which the algorithm of Section 3 accomplishes with essentially no increase in computational complexity as compared to the processing of single scale data. Finally in Section 4.3 we examine a $1-D$ version of the computer vision problem of motion estimation, illustrating that the use of a slight variation on standard formulations, involving a "fractal prior" obtained from a tree model, yields excellent results while offering substantial computational advantages. In particular, while all of the examples in this section are $1-D$, they all can be extended to $2-D$, in which the potential computational savings can be quite dramatic.

4.1 Smoothing Gauss-Markov and 1/f Processes

In this section we consider smoothing problems for two stochastic processes, neither of which exactly fits the tree model used in this paper. The first of these is a discrete-time first-order Gauss-Markov process, *i.e.* a stationary time series given by the standard first-order differnce equation

$$x_{n+1} = \alpha x_n + w_n \tag{4.1}$$

where, for simplicity, we normalize the variance of x_n to a value of 1, so that the variance of the white noise sequence w_n is $(1-\alpha^2)$. For our example we take $\alpha = .9006$, a value arrived at by sampling and aliasing considerations [42]. We consider the following measurements of x_n.

$$y_n = x_n + v_n, \ \text{n=0,\ldots,}N-1 \tag{4.2}$$

where v_n has variance R so that the SNR in (4.2) is $R^{-\frac{1}{2}}$. In the examples that follow we take $N = 128$.

As developed in [7,11,17] while the wavelet transform of $\{x_n | 0 \leq n \leq N-1\}$ does not yield a completely whitened sequence of wavelet coefficients, it does accomplish a substantial level of decorrelation, —*i.e.* the detail added to the approximation at each scale is only weakly correlated with the coarser–scale approximation. Furthermore, decorrelation is improved by using wavelets of larger support—and in particular those with increasing numbers of vanishing moments [7,14] —, suggesting a wavelet-transform-based estimation algorithm developed in [11,42]. It also provides the motivation for the example presented here in which we smooth the data in (4.2) using a model for x_n as the finest level process generated by a scalar model of the form (2.1), (2.2), where in this case the data (4.2) corresponds to measurements only at the finest level of the tree. In particular we have considered two scalar tree models, the first being a constant parameter model in steady-state

$$x(t) = ax(t\bar{\gamma}) + w(t) \tag{4.3}$$

with $w(t)$ having variance equal to $p(1-a^2)$, where p is the variance of $x(t)$. In our example in which the length of the signal x_n is 128, the model (4.3) evolves from the root node to level $m = 7$, at which we obtain our approximate model for x_n and at which we incorporate the measurements (4.2).

A second model that we considered has the property that its fine-scale variations have smaller variance than its coarser-scale variations. In particular, as developed in [16,32,36], fractal-like processes which have spectra that vary as $1/f^{\beta}$ for some value of β, have wavelet variances that decrease geometrically with m. Thus we consider a variation on the model (4.3):

$$x(t) = ax(t\overline{\gamma}) + 2^{-\frac{\delta m(t)}{2}} w(t) \tag{4.4}$$

where δ controls the scale-to-scale geometric decay in the variance of the noise term, $w(t)$ has unit variance, and the variance of the initial condition, $x(0)$, at the top of the tree is p_0.

Models of this type were used to design suboptimal estimators for the Gauss-Markov process (4.1) based on the data (4.2) at several different SNR's. Figure 4a illustrates a sample path of (4.1) and data (4.2) at $SNR = \sqrt{2}$, while Figure 4b compares the Gauss-Markov sample path to the optimal estimate using a standard optimal smoothing algorithm for the model (4.1). To compare this optimal performance to that of algorithms based on the tree models (4.3), (4.4) we define a metric for comparison based on error variance reduction. Specifically, if p is the prior variance of a process, and p_s is the error covariance of any particular smoothing algorithm, then the variance reduction percentage achieved by this smoother is

$$\rho = \frac{p_o - p_s}{p_o} \tag{4.5}$$

Then if ρ_o denotes the value of ρ for the optimal smoother and ρ_s denotes the value for a suboptimal estimate, we define the following performance degradation measure:

$$\Delta = \frac{\rho_o - \rho_s}{p_o} \tag{4.6}$$

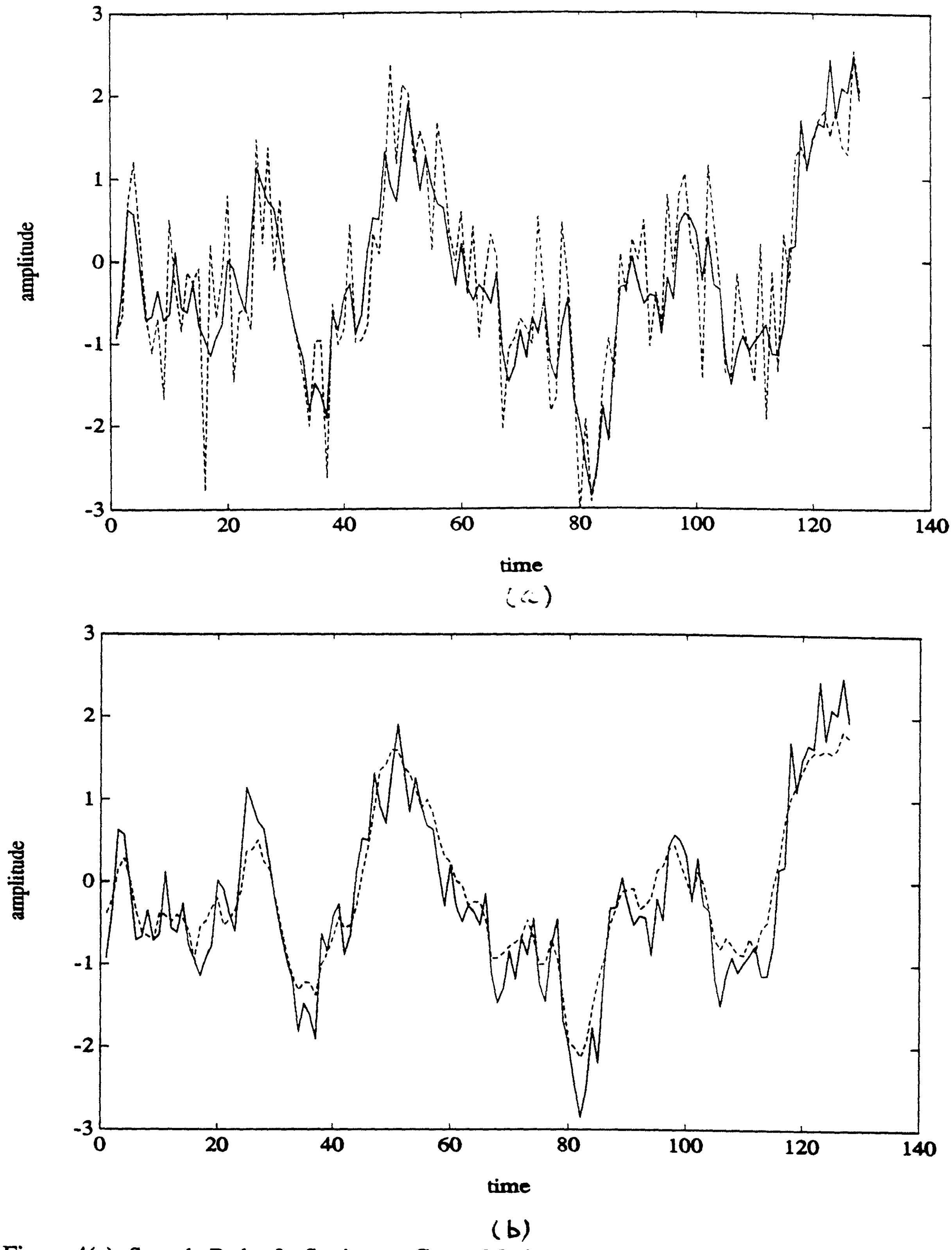


Figure 4(a): Sample Path of a Stationary Gauss-Markov Process (solid) and Its Noisy Measurement with SNR = 1.4142 (dashed); (b) the same Gauss-Makov sample path (solid) and its smoothed version using the optimal smoother for the Gauss-Markov model.

	Model (4.3)	Model (4.4)
$SNR = 2.8284$	1.11 %	1.08 %
$SNR = 1.412$	3.55 %	3.31 %
$SNR = .7071$	7.59 %	6.88 %
$SNR = .5$	10.52 %	9.15 %

Table 1: Performance Degradation Comparison for Tree Smoothers Based on Models (4.3) and (4.4)

For each SNR considered and for each of the models (4.3), (4.4) we chose the model parameters (a and p for (4.3) and a , ρ_o, and δ for (4.4)), to minimize the corresponding value of Δ. Table 1 presents the resulting values of Δ, expressed in percentage form, indicating that estimators based on these models achieve variance reduction levels within a few percent of the optimal smoother. [4] This is further illustrated in Figure 5 in which we compare in (a) the optimal estimate at $SNR = \sqrt{2}$ for a tree smoother based on the model (4.4) with the original sample path. In Figure 5b we compare the estimates produced by the optimal filter (Fig.4b) and that of Fig.5a, providing further evidence of the excellent performance of our estimators. One should, of course, ask why this is significant, since we already have optimal smoothing algorithms for Gauss-Markov processes. There are at least three reasons: (1) the tree algorithms, with their pyramidal structure, are highly parallelizable; (2) as we will see in the next section, these algorithms directly allow us to fuse multiresolution data; and (3) perhaps most importantly, these same ideas extend to $2-D$ data where the computational savings over previously known algorithms should be substantial.

As a second example we consider smoothing for a $1/f$-like fractal process of the type developed in [36]. Specifically as shown in [36], we can use the synthesis form of the wavelet transform to construct processes possessing self-similar statistical properties and with spectra which are nearly $1/f$ by taking the wavelet coefficients to be

[4] Note that the increasing size of Δ with decreasing SNR is due for the most part to the decrease in the denominator of (4.6), *i.e.* at low SNR's only minimal variance reduction is achieved by any estimator.

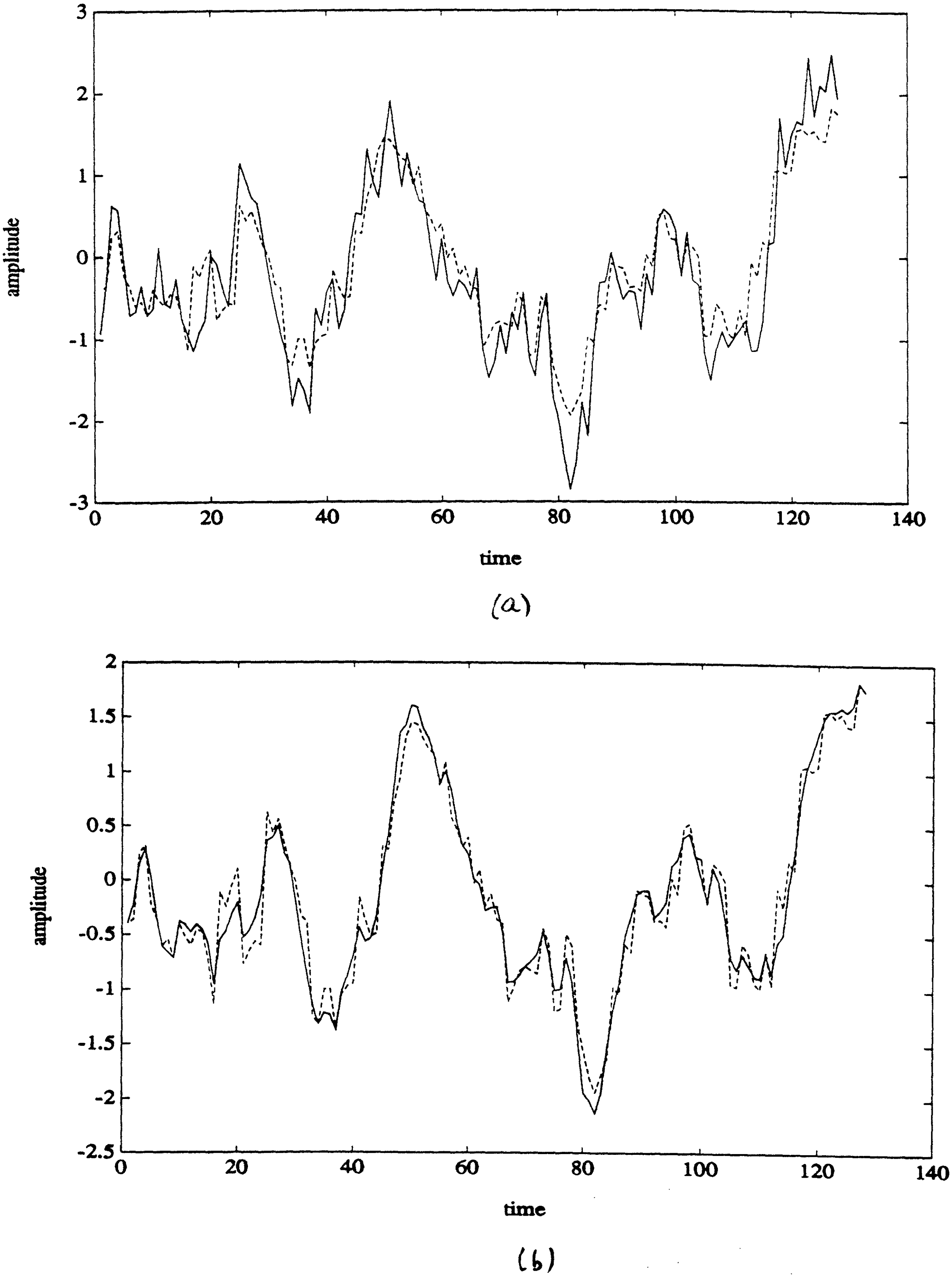


Figure 5: (a) Stationary Gauss-Markov process (solid) versus the result of smoothing of the noisy data of Fig. 4a (dashed) based on the model (4.4) with parameters a=.9464, b=1, p_o=7.7462, δ=.5059; and (b) comparison of the optimal smoothing estimate of Fig. 4a (solid) versus the estimate in Fig. 5a (dashed)

white and with variances that decay geometrically at finer scales. In Figure 6a we illustrate the sample path of such a process using the 4-tap wavelet filter of Daubechies [14], as well as noisy data with an SNR of $\sqrt{2}$. In this case the scale-dependent geometric decay in wavelet coefficient variances is 2^{-m}, yielding a $1/f^{\beta}$ spectrum with $\beta = 1$. In Figure 6b we illustrate the optimal estimate in this case. As developed in [11,37] this optimal estimator can be implemented by taking the (4-tap) wavelet transform of the data and then filtering each wavelet coefficient separately (taking advantage of the fact that the wavelet transform yields the Karhunen-Loeve expansion for the data). In Figure 7a we illustrate the result of smoothing the noisy data of Figure 6a based on a tree model of the form of (4.4), and in Figure 7b we compare this estimate to the optimal estimate of Figure 6b. The average performance degradation of the tree smoother is only 2.76%, although, as indicated in Figure 7, the tree smoother appears to do *better* over certain portions of the signal, indicating that our models are well-suited to capturing and tracking $1/f$-like behavior. Again given the existence of effective optimal algorithms [11,37], there is the natural question of why we should bother with the tree-based algorithm in this case, especially since these methods are very efficient, the methods of [11] can be applied to multiresolution data, and they can be extended to $2-D$. The reason is that, for these wavelet-transform-based methods to apply, it is necessary that the identical type and quality of measurement be available along each scale (i.e. $C(t)$ in (2.2) must only depend on $m(t)$). In the next two sections we encounter important examples in which this is not the case, and thus the methods of [11,37] are not applicable, but our tree-based methods are.

4.2 Multiresolution Data Fusion

In this section we illustrate the application of our tree-based smoothing algorithms to the fusion of multiresolution data. While we could provide examples in which data are available at all nodes at several levels of the tree— and such examples are presented in [11,42] using wavelet-transform-based algorithms— we focus here on examples to which neither standard smoothing algorithms nor wavelet-based methods apply. In particular we consider an example in which we wish to fuse fine-scale data of limited

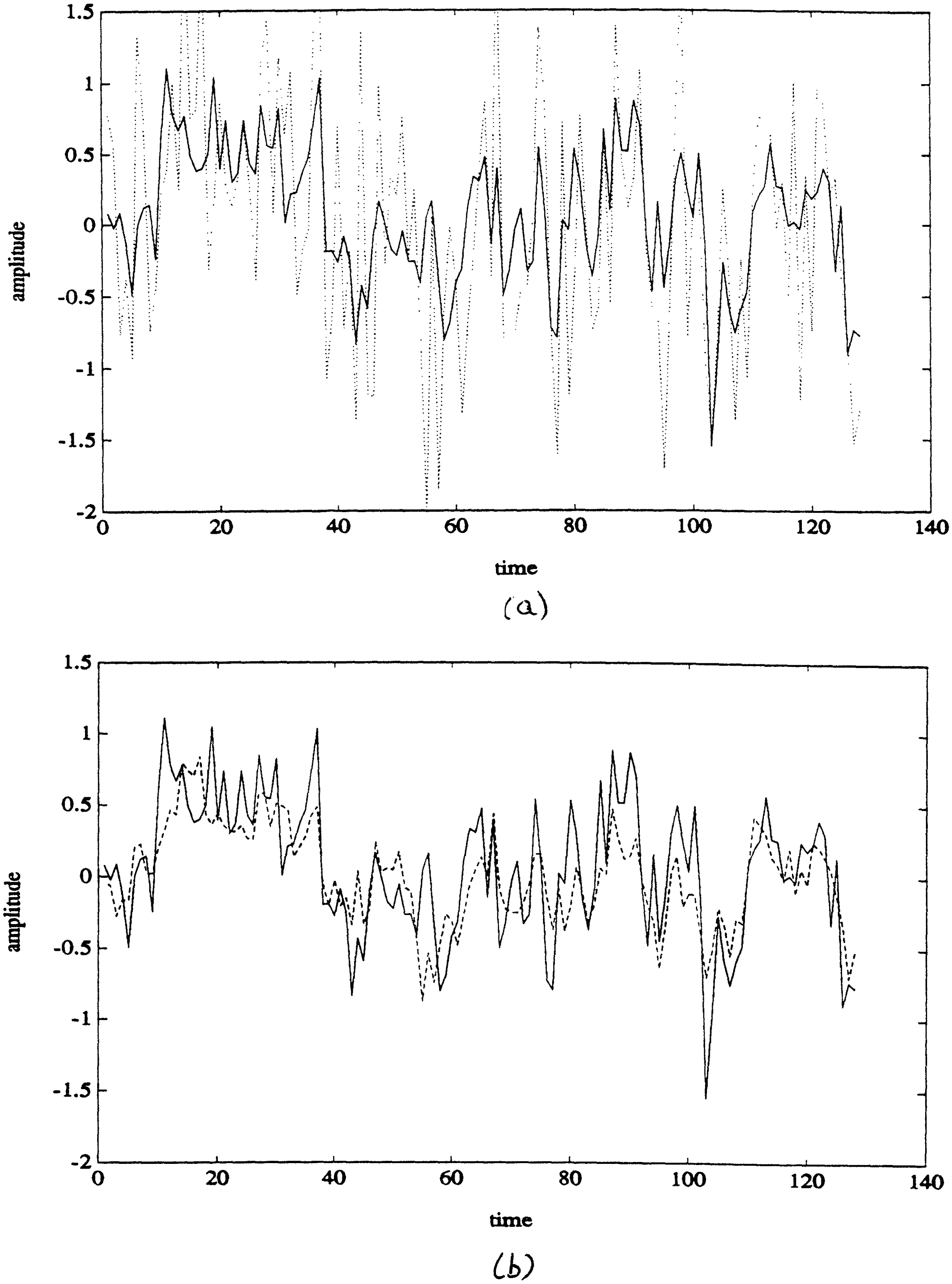
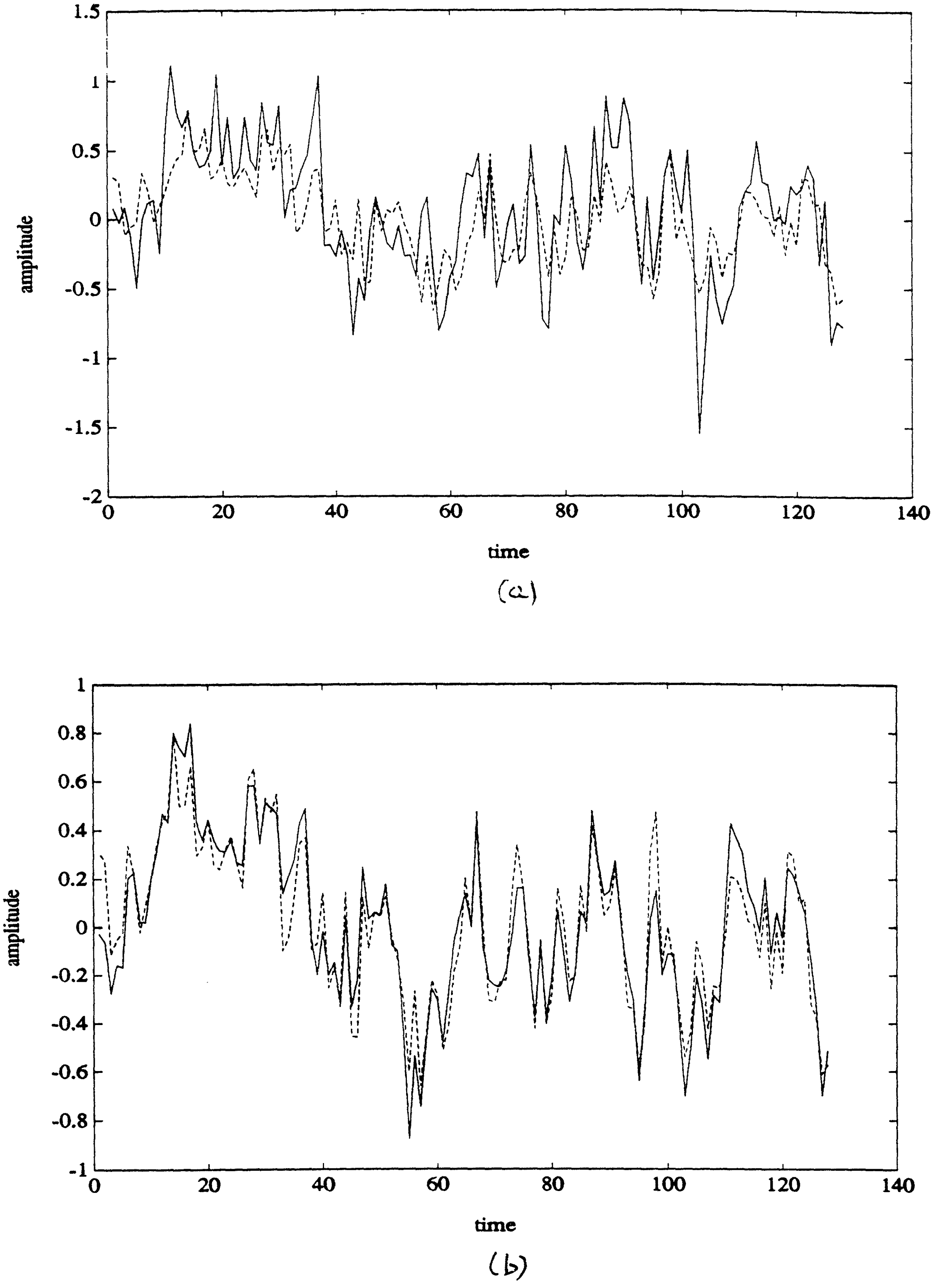


Figure 6: (a) A 1/f process (solid) generated using the method of [36] and its noisy measurement (dotted) with SNR=$\sqrt{2}$; and (b) comparison of the 1/f sample path (solid) versus the optimal estimate (dashed) based on the noisy data of part (a).



(a)

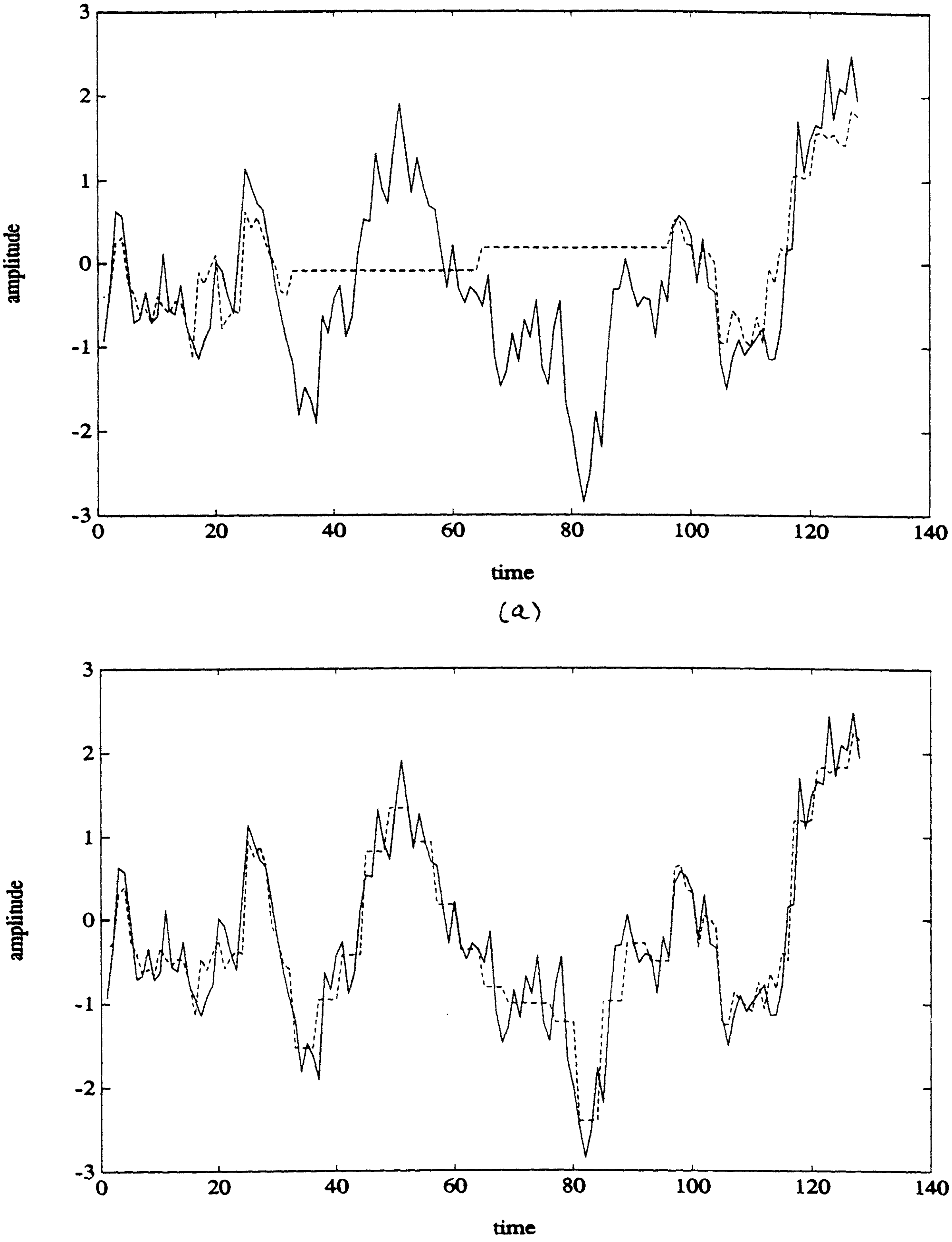
(b)

Figure 7: (a) The 1/f sample path of Fig. 6 versus the optimal estimate based on the model (4.4) with a=.6694, b=1, p_0=1.0131, δ=.4843; and (b) comparison of the optimal estimates of Figures 6b and 7a.

coverage with full coverage coarse-scale data. Problems of this type arise, for example, in atmospheric or oceanographic sensing in which both broad coverage satellite data and sparse point measurement data (e.g. from surface measurement stations) are available.

Consider first the Gauss-Markov process of Figure 4a where we assume that only 50% of the data in this figure are available, corresponding to the 32 data points at each end of the data interval. In Figure 8a we illustrate the optimal tree-smoothed estimate based on the model of (4.4) and using only this limited data set. In contrast, in Figure 8b we display the optimal tree-smoothed estimate based on the fusion of the limited fine-scale data with a full set of high quality measurements at a level two scales coarser (i.e. where there are 32 points rather than 128). Note that in addition to providing information that guides the interpolation of the fine-scale data, the coarse data also provides information that allows us to remove offset errors in regions in which noisy fine scale data are available. By using the multiscale covariance equations of Section 3 we can quantify the value of such coarse data, as illustrated in Figure 9 in which we display the variation of performance with the level at which the coarse data is available. We refer the reader to [42] for other examples, including for the $1/f$ process of Figure 6.

Finally, let us comment on the measurement model for the coarse-level data in the example depicted in Figure 8b. In particular, the coarse data used in the example of Figure 8 were noise-corrupted measurements of 4-point averages of the fine-level process x_n. Note that using the model (4.4) we can view these as measurements two levels higher in the tree. For example, from (4.4) it is straightforward to see that the average of $x(t\alpha^2)$, $x(t\alpha\beta)$, $x(t\beta\alpha)$ and $x(t\beta^2)$ is a scaled version of $x(t)$ corrupted by the noises $w(t\alpha)$, $w(t\beta)$, $w(t\alpha^2)$, $w(t\alpha\beta)$, $w(t\beta\alpha)$, and $w(t\beta^2)$. From this it is straightforward to obtain a model for the coarse measurements in which the measurement noise also reflects finer-scale process fluctuations captured by the w's. Note that to be precise, this noise is correlated with the finer level data, and thus a truly optimal tree-based smoother would need to account for this via state augmentation. However the result presented in Figure 8b was generated using a tree-



(a)

(b)

Figure 8: (a) Sample path of the Stationary Gauss-Markov Process (solid) of Fig. 4a and the Result of Using the Tree Smoother based on (4.4) and using Sparse Data of SNR=1.4142 (dashed); (b) comparing the Gauss-Markov sample path (solid) with the result (dashed) of using the tree smoother to fuse high quality coarse data (at two scales above the finest scale) of SNR=100 with the sparse fine data of SNR=1.4142.

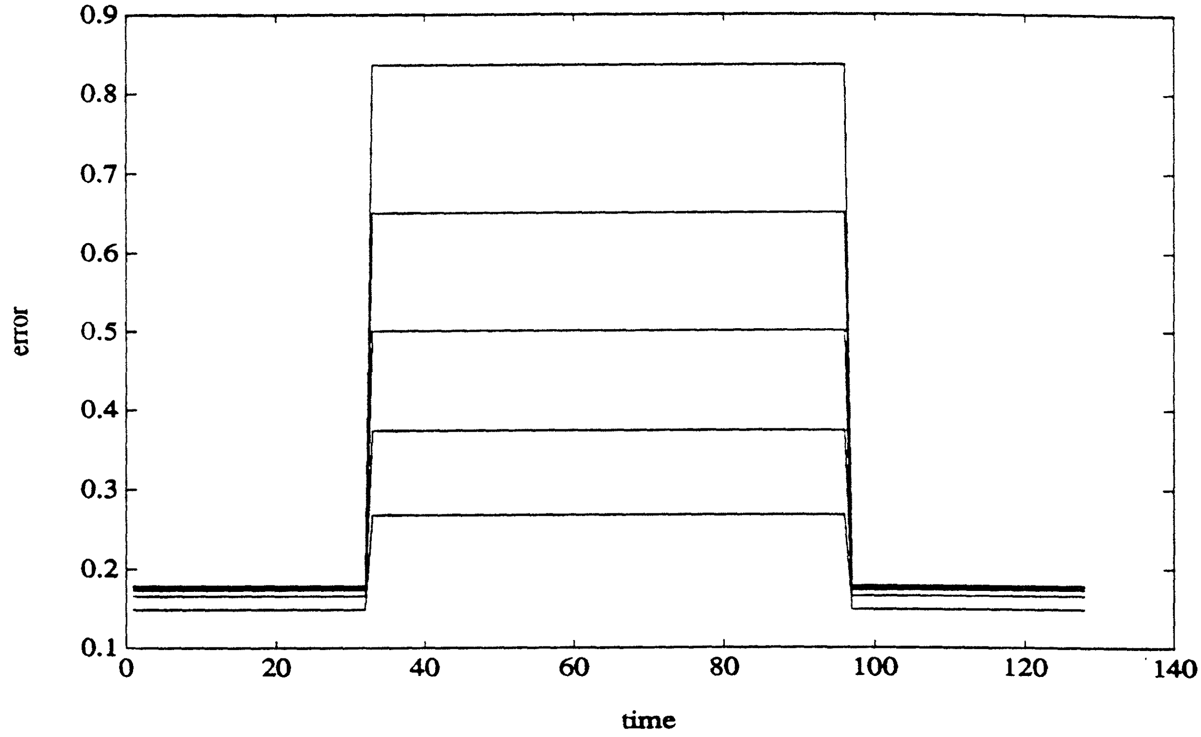


Figure 9: Plots of Performance for the Case of Full Coverage Coarse Data Fused with Sparse Fine Data Both of SNR=1.4142. Five Plots Correspond to Coarse Data 1) One Level Coarser Than Fine Data 2) Two Levels Coarser... 5) Five Levels Coarser

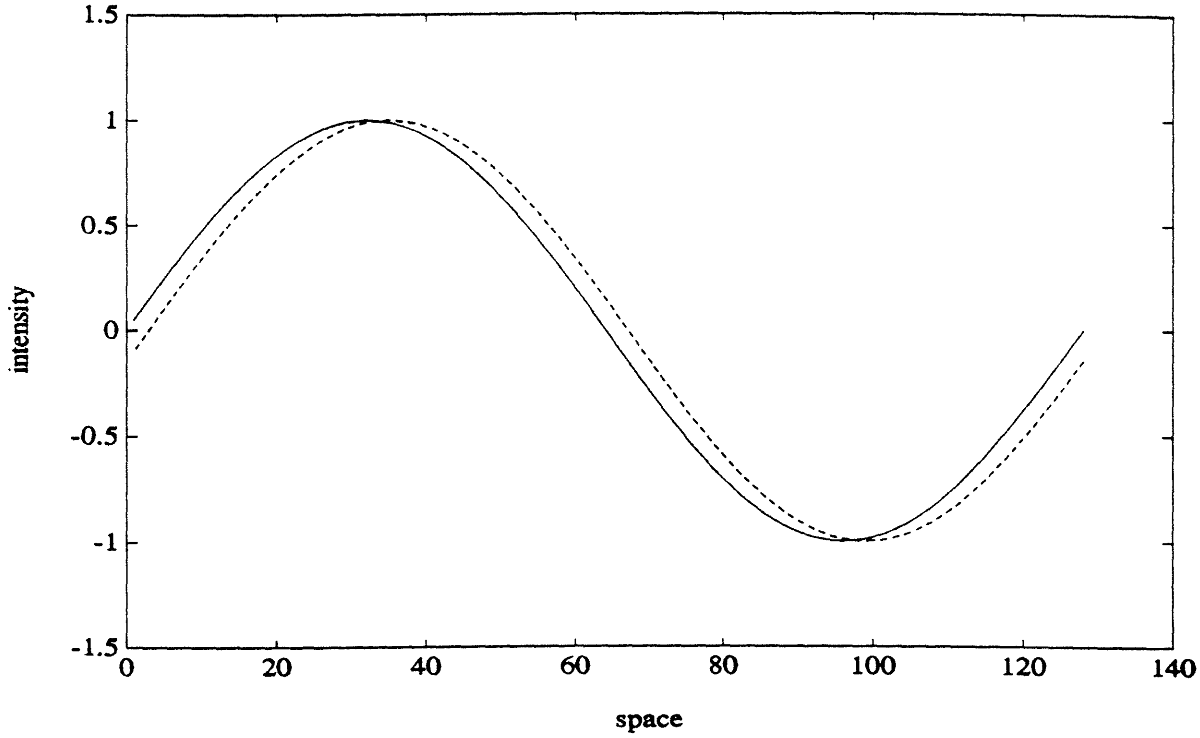
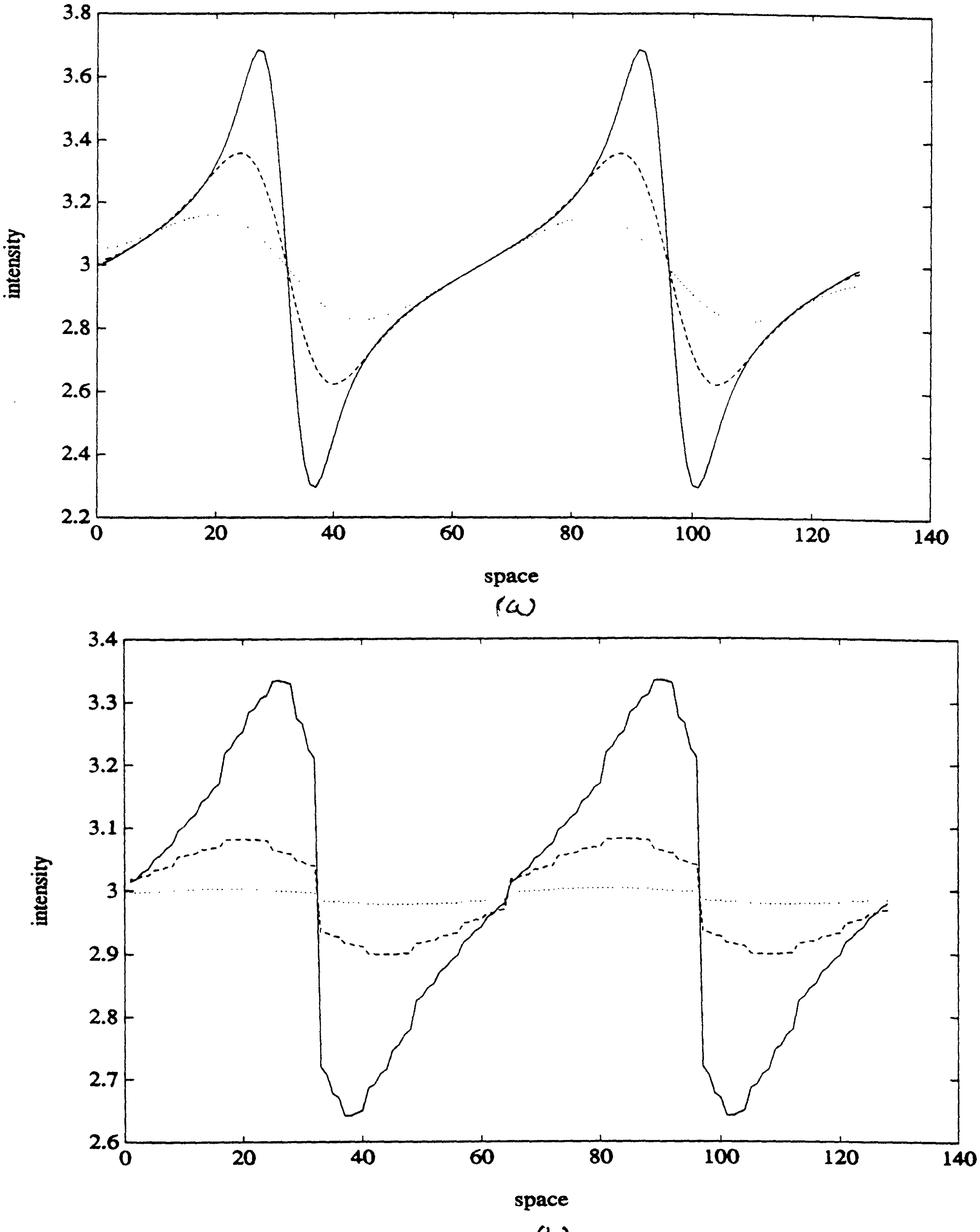


Figure 10: Translating Sinusoid at Time t (solid) and at Time t+1 (dashed)



(a)

(b)

Figure 11: (a) Estimate of v(x) Using Standard Regularization for Γ=1 (solid), for Γ=.1 (dashed), and for Γ=.01 (dotted); and (b) Estimate of v(x) using the tree smoother for Γ=1 (solid), Γ=.1 (dashed), and Γ= .01 (dotted)

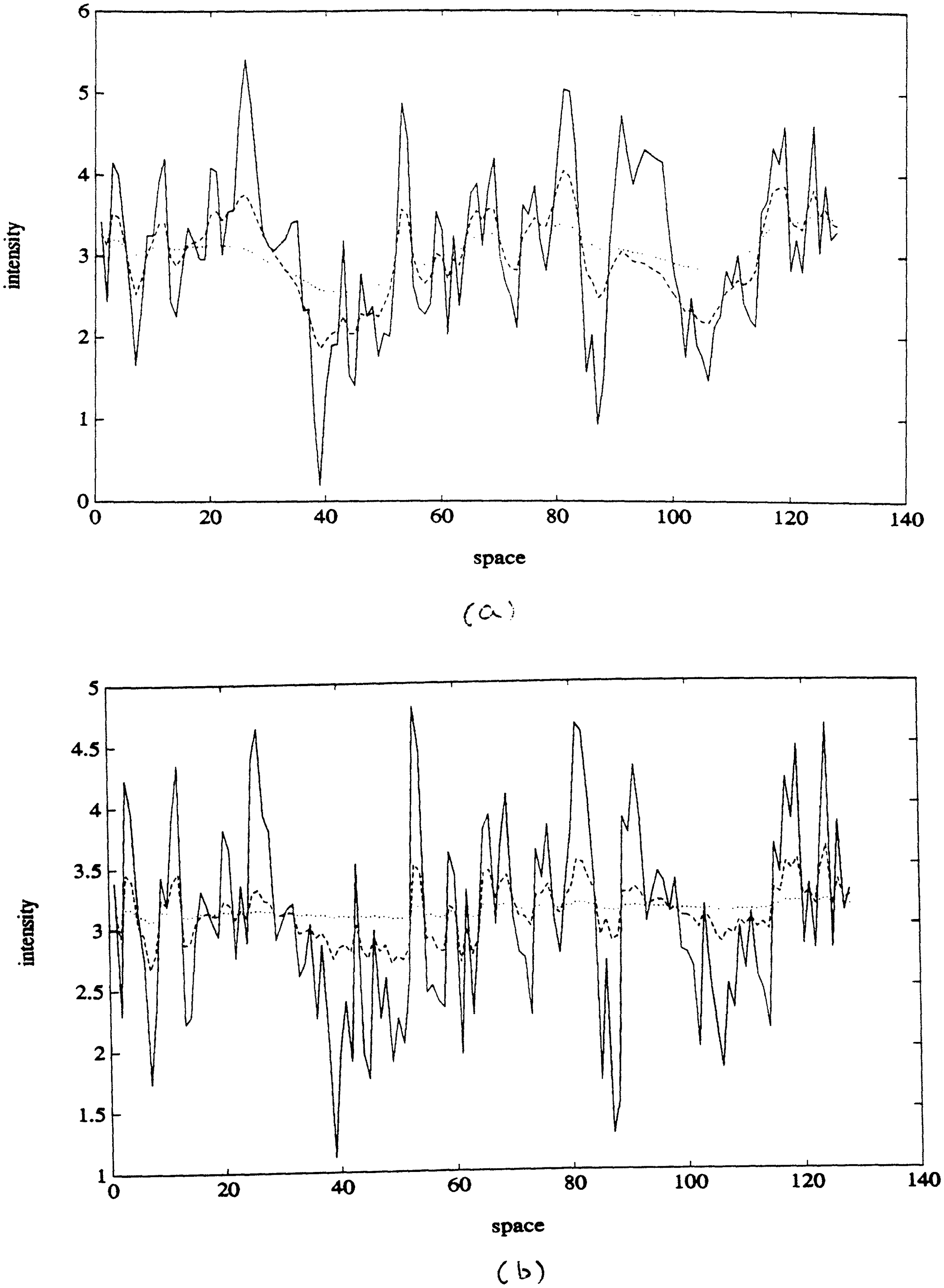


Figure 12: (a) Estimate of v(x) for the translating sinusoid based on noisy data and Using Standard Regularization for Γ=1 (solid), for for Γ=.1 (dashed), and for Γ=.01 (dotted); (b) Analogous estimates using the tree smoother.

based smoother that neglected this correlation. As the quality of the result in the figure indicates, excellent performance is achieved with this simplification.

4.3 Motion Estimation

In this section we illustrate the use of our tree model in order to provide statistical regularization for a $1-D$ version of a representative problem arising in computer vision, namely that of estimating *optical flow*, i.e., motion in the image frame that is apparent from observation of the temporal fluctuations in the intensity patterns in an image sequence. Regularization methods are common in many computer vision problems such as optical flow estimation and in other contexts including many inverse problems of mathematical physics, and the approach described in this section should prove to be of considerable value in many of these as well as in full $2-D$ and $3-D$ image analysis problems. In particular the usual "smoothness" regularization methods employed in many problems lead to computationally intensive variational problems solvable only by iterative methods. By adoptimg a statistical perspective and by modifying the regularization slightly, we obtain a formulation exactly of the form of our tree smoothing problem, leading to non-iterative, highly parallelizable and efficient scale-recursive solutions.

The results in this section focus on a 1D version of the problem as formulated in Horn and Schunck[19]. Let $f(x,t)$ denote a 1-D "image sequence", i.e. for each t, $f(x,t)$ is a function of the 1-D spatial variable x, representing image intensity variations with space. The constraint typically used in gradient-based approaches to motion estimation is referred to in the literature as the "brightness constraint"[19], and it amounts to assuming that the brightness of each point in the image, as we follow its movement, is constant. In other words, at each time t the total derivative of the image intensity is zero, i.e.

$$\frac{Df}{dt} = 0 \tag{4.7}$$

This equation can be rewritten as follows.

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} = 0 \tag{4.8}$$

The quantity $\frac{\partial x}{\partial t}$ is referred to as the *optical flow*, which we denote as the spatial function, $v(x)$, and it is this quantity which we would like to determine from observing $f(x,t)$ sequentially in time. That is, at each t we assume that we can extract measurements of the temporal and spatial derivatives of $f(x,t)$, and we then want to use these, together with (4.8), to estimate the velocity field $v(x)$ at this instant in time.

We rewrite eq.(4.8) as

$$y(x) = c(x)v(x) \tag{4.9}$$

$$y(x) = -\frac{\partial f}{\partial t} \tag{4.10}$$

$$c(x) = \frac{\partial f}{\partial x} \tag{4.11}$$

The following is the optimization problem formulation of Horn and Schunk [19] for determining $v(x)$.

$$\hat{v}(x) =^{\overset{argmin}{v}} \int \{\mu\|y(x) - c(x)v(x)\|^2 + \|\frac{dv}{dx}\|^2\}dx \tag{4.12}$$

The second term in the cost function, $\|\frac{dv}{dx}\|^2$, is referred to as the "smoothness" constraint as it is meant to penalize large derivatives of the optical flow, constraining the solution to have a certain degree of smoothness. In the standard 2-D image processing problem $v(x)$ is a 2-D velocity vector field, $\frac{\partial f}{\partial x}$ is the 2-D gradient of image intensity, and, the brightness constraint (4.8) provides a scalar constraint at each spatial location for the 2-D vector $v(x)$. In this case the optical flow problem is ill-posed, and the smoothness penalty makes it well-posed by, in essence, providing another (noisy) constraint in 2-D. The other purposes of the smoothness constraint is to reduce the effects of noise in the measurement of the temporal and spatial gradients and to allow interpolation of motion over regions of little or no contrast, i.e. regions where $c(x)$ is zero or nearly zero. In our 1-D problem, we do not have the issue of ill-posedness, so that the role of the smoothness constraint is solely for the purposes of noise rejection and interpolation in areas of zero or near-zero intensity variations. Note that computing $\hat{v}(x)$ in (4.12) is potentially daunting as the dimension of $\hat{v}(x)$ is equal to the number of pixels in the image.

As in Rougee, et al [27] the optimization problem in (4.12) can be interpreted as a stochastic estimation problem. In particular the smoothness constraint can be interpreted as the following prior model on $v(x)$.

$$\frac{dv}{dx} = w(x) \tag{4.13}$$

where $w(x)$ is a 1-D white noise process with unit intensity. The estimation problem equivalent to (4.12) consists of estimating $v(x)$ based on the following observation equation.

$$y(x) = c(x)v(x) + r(x) \tag{4.14}$$

where $r(x)$ is white noise with intensity μ^{-1}. Henceforth, we refer to (4.13),(4.14) as the *standard model.*

Let us now examine the particular prior model (4.13) associated with the smoothness constraint. In particular we see that $v(x)$ is in fact a *Brownian motion process*, i.e. a self-similar, fractal process with a $1/f^2$ spectrum. For this reason (4.12) is sometimes referred to as a "fractal prior". Given that the introduction of this prior is simply for the purposes of regularization, we are led directly to the idea of replacing the smoothness constraint model (4.13) by one of our tree models. Since in 2-D, solution of (4.12) corresponds to solving coupled partial differential equations [19], the use of a tree model and the resulting scale-recursive algorithm offer the possibility of substantial computational savings.

Let us consider a simplified, discretized, version of this problem, i.e. where $f(x,t)$ is observed only at integer values of x and t. In this case we need to approximate $\frac{\partial f}{\partial t}$ and $\frac{\partial f}{\partial x}$. For simplicity in our discussion, we consider the following finite difference approximations of these partial derivatives.

$$\frac{\partial f}{\partial t}|_{x,t} \approx f(x,t+1) - f(x,t) \tag{4.15}$$

$$\frac{\partial f}{\partial x}|_{x,t} \approx (f(x+1,t) - f(x-1,t))/2 \tag{4.16}$$

Obviously these rather crude approximations will lead to distortions (which we will see here), and more sophisticated methods can be used. In particular, as discussed in [41], the highest velocity that can be estimated depends both on the temporal sampling rate and the spatial frequency content of $f(x,t)$. In particular, low-pass spatial filtering of $f(x,t)$ not only is useful for the reduction in noise in the derivatives (4.15), (4.16) but also in allowing the estimation of larger inter-frame spatial displacements via the differential brightness constraint (4.8). As we are simply interested in demonstrating the promise of our alternate formulation, we confine ourselves here to the simple approximation (4.15), (4.16).

We assume that our image is available at two time instants, t and $t+1$, where each image is uniformly sampled over a finite interval in space; i.e. we have $f(i,t)$ and $f(i,t+1)$ where $i \in \{0,1,2,...N-1\}$. The discretized smoothing problem for our standard model is as follows.

$$\begin{aligned} v(i+1)-v(i) &= w(i) && (4.17)\\ y(i) &= c(i)v(i)+r(i) && (4.18)\\ c(i) &= f(i+1,t)-f(i,t) && (4.19) \end{aligned}$$

where the white noises $w(i)$ and $v(i)$ have intensities 1 and μ^{-1}, respectively. The solution to this smoothing problem is given by

$$\hat{v} = (\mathcal{L} + \mu^{-1}C^TC)^{-1}(\mu^{-1}C^T)\overline{y} \tag{4.20}$$

where $\hat{v}$ is the vector of velocity estimates for all i, $\overline{y}$ is the vector of measurements, and

$$C = \begin{bmatrix} c(0) & 0 & \cdots & \cdots & 0 \\ 0 & c(1) & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & c(N-3) & 0 \\ 0 & \cdots & & 0 & c(N-2) \end{bmatrix} \tag{4.21}$$

and

$$\mathcal{L} = \begin{bmatrix} 1 & -1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & \cdots & 0 \\ & & & \ddots & \ddots & \ddots & \\ \vdots & & & & & & \vdots \\ 0 & \cdots & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & & 0 & -1 & 1 \end{bmatrix} \tag{4.22}$$

Note that the matrix $\mathcal{L}$, which is a discrete approximation of the operator $\frac{d^2}{dx^2}$, is a result of the prior model we have chosen.

The multiscale model we use as our prior model for $v(x)$, as an alternative to our discretized standard model, is the tree model, (4.4) where t indexes the nodes of a finite tree with N points at the bottom level. Thus, the covariance $P(\theta)$ of this zero-mean process at the bottom level of the tree is specified entirely by the parameter vector $\theta = [a, p_0, \gamma, b]$. To choose the parameters of our tree model so as to yield an approximation to the standard model, we fit the information matrix of our tree process, $P^{-1}(\theta)$, to $\mathcal{L}$ by minimizing the matrix 2-norm of the difference between $\mathcal{L}$ and $P^{-1}(\theta)$.

$$\theta_{fitted} = \overset{argmin}{\theta} \; |\mathcal{L} - P^{-1}(\theta)|_2 \tag{4.23}$$

In comparing the performance of the multiscale smoother with the performance of the standard regularization method, we need a way of normalizing the problem. We define the following quantity, which can be thought of as the ratio between the information due to measurements and the information due to the model.

$$\Gamma \triangleq \frac{trace(\mu C^T C)}{trace(\mathcal{I})} \tag{4.24}$$

where $\mathcal{I}$ is either $\mathcal{L}$ or $P^{-1}(\hat{\theta})$. For our examples, we vary Γ by varying μ,

Figure 10 shows snapshots of the image of a translating sinusoid at times t and $t+1$, while Figure 11a shows the result of estimating v based on standard regularization for $\Gamma = 1, .1, .01$, and Figure 11b shows the result of estimating v based on our tree smoother for $\Gamma = 1, .1, .01$. The true value of v is a constant equal to 3. The substantial deviations from the value of 3 are due to the inaccuracy of the approximations (4.15), (4.16). Note that the two approaches yield similar results. In fact for all three values of Γ our tree smoother actually performs better. As we would expect by decreasing Γ, i.e. decreasing the weight μ of the measurment term in the cost function, the effect of the approximation error in (4.15), (4.16) is reduced, and smoother estimates result. Figures 12a and b show similar behavior when the data of Figure 10 is observed in noise. Again the performance is somewhat better for the tree model, but this is not our main point. Rather, what we have shown is that *comparable* results can be obtained with tree-model-based algorithms, and, given their considerable computational advantages, they offer an attractive alternative worthy of further development.

5 Conclusions

In this paper we have developed a new method for the multiresolution modeling of stochastic processes based on describing their scale-to-scale construction using dynamic models defined on dyadic trees. As we have seen, this framework allows us to describe a rich class of stochastic processes and also leads to an extremely efficient and highly parallelizable scale-recursive optimal estimation algorithm generalizing the Rauch-Tung-Striebel smoothing algorithm to the dyadic tree. This algorithm involves a variation on the Kalman filtering algorithm in that, in addition to the usual measurement update and (fine-to-coarse) prediction steps, there is also a data fusion step. This in turn leads to a new Riccati equation, which we analyze in detail in [43] using several system-theoretic concepts for systems on trees. We have also illustrated the potential of this ııamework in providing highly parallel algorithms for the smoothing of broad classes of stochastic processes, for the fusion of multiresolution data, and for the efficient solution of statistically regularized problems such as arise in computer vision.

We believe that this framework has considerable promise, and numerous directions for further work suggest themselves. In particular the extension and application of these ideas in 2-D offers numerous possibilities such as for the motion estimation problem described in Section 4. Also, there are several reasons to believe that our tree models can be used to describe a surprisingly rich class of stochastic processes. For example, a recent extension of wavelet transforms is the class of so-called wave packet transforms [13] in which both the fine resolution and coarse resolution features are subject to further decomposition. These correspond, in essence, to higher-order models in scale and thus correspond to higher-dimensional versions of our state models. These and a variety of other examples offer intriguing possibilities for future investigation.

A Appendix

In this appendix we verify the formulae (3.15), (3.16) for the fusion of the estimates $\hat{x}(t|t\alpha)$ and $\hat{x}(t|t\beta)$ to produce $\hat{x}(t|t+)$. By definition

$$\hat{x}(t|t+) = E[x(t)|Y_{t\alpha}, Y_{t\beta}] \tag{A.1}$$

But from our model (2.1),(2.2) we can decompose $Y_{t\alpha}$ and $Y_{t\beta}$ in the following way.

$$Y_{t\alpha} = M_{t\alpha}x(t) + \xi_1 \tag{A.2}$$

$$Y_{t\beta} = M_{t\beta}x(t) + \xi_2 \tag{A.3}$$

where the matrices $M_{t\alpha}$ and $M_{t\beta}$ contain products of $A(s)$, $m(s) > m(t)$, and the vectors ξ_1 and ξ_2 are functions of the driving noises $w(s)$ and the measurement noises $v(s)$ for s in the subtrees strictly below $t\alpha$ and $t\beta$, respectively, the latter fact implying $\xi_1 \perp \xi_2$. Let $R_{t\alpha}$ and $R_{t\beta}$ denote the covariances of ξ_1 and ξ_2, respectively. Then, rewriting (A.2), (A.3) as

$$\mathcal{Y} = \mathcal{H}x(t) + \Xi \tag{A.4}$$

where

$$\mathcal{H} = \begin{bmatrix} M_{t\alpha} \\ M_{t\beta} \end{bmatrix}, \Xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}, \mathcal{R} = E[\Xi\Xi^T] \tag{A.5}$$

and $x(t) \perp \Xi$, we can write the optimal estimate of $x(t)$ given $\mathcal{Y}$ in the following way:

$$\begin{aligned} P(t|t+) &= [P_x^{-1}(t) + \mathcal{H}^T\mathcal{R}^{-1}\mathcal{H}]^{-1} \\ &= [P_x^{-1}(t) + M_{t\alpha}^T R_{t\alpha}^{-1} M_{t\alpha} + M_{t\beta}^T R_{t\beta}^{-1} M_{t\beta}]^{-1} \end{aligned} \tag{A.6}$$

$$\hat{x}(t|t+) = P(t|t+)\mathcal{H}^+\mathcal{R}^{-1}\mathcal{Y} = P(t|t+)[M_{t\alpha}^T R_{t\alpha}^{-1} Y_{t\alpha} + M_{t\beta}^T R_{t\beta}^{-1} Y_{t\beta}] \tag{A.7}$$

Similarly

$$\hat{x}(t|t\alpha) = P(t|t\alpha) M_{t\alpha}^T R_{t\alpha}^{-1} Y_{t\alpha} \tag{A.8}$$

where

$$P(t|t\alpha) = [P_x^{-1}(t) + M_{t\alpha}^T R_{t\alpha}^{-1} M_{t\alpha}]^{-1} \tag{A.9}$$

and analogous equations hold for $\hat{x}(t|t\beta)$ and $P(t|t\beta)$. Eqs. (3.15), (3.16) then follow immediately from (A.6)-(A.9).

B Appendix

In this appendix we verify the RTS smoothing recursions (3.17)-(3.19). The key to this is the following orthogonal decomposition of Y_0 (i.e. the full set of *all* measurements at every node on the tree). Specifically, for each t, Y_t, is given by (3.1), and we let $Y_{\bar{t}}$ denote all the remaining measurements. Viewing these as vectors we define

$$\nu_{\bar{t}|t} = Y_{\bar{t}} - E[Y_{\bar{t}}|Y_t] \tag{B.1}$$

so that $\nu_{\bar{t}|t} \perp Y_t$ and the linear span of the set of all measurements, is given by

$$\text{span } Y_0 = \text{ span } \{Y_t, Y_{\bar{t}}\} = \text{ span } \left\{Y_t, \nu_{\bar{t}|t}\right\} \tag{B.2}$$

Then

$$\begin{aligned} \hat{x}_s(t) &= E[x(t)|Y_t, \nu_{\bar{t}|t}] \\ &= \hat{x}(t|t) + E[x(t)|\nu_{\bar{t}|t}] \end{aligned} \tag{B.3}$$

If we write $x(t)$ as

$$x(t) = \tilde{x}(t|t) + \hat{x}(t|t) \tag{B.4}$$

and note that

$$\hat{x}(t|t) \perp \nu_{\bar{t}|t} \tag{B.5}$$

then we can write the following.

$$\hat{x}_s(t) = \hat{x}(t|t) + E[\tilde{x}(t|t)|\nu_{\bar{t}|t}] \tag{B.6}$$

Using the same argument on $\hat{x}_s(t\bar{\gamma})$ allows us to write

$$\hat{x}_s(t\bar{\gamma}) = \hat{x}(t\bar{\gamma}|t) + E[\tilde{x}(t\bar{\gamma}|t)|\nu_{\bar{t}|t}] \tag{B.7}$$

Suppose the following equality were to hold.

$$E[\tilde{x}(t|t)|\nu_{\bar{t}|t}] = L(E[\tilde{x}(t\bar{\gamma}|t)|\nu_{\bar{t}|t}]) \tag{B.8}$$

where L is a matrix. Then (B.6) and (B.7) could be combined to yield

$$\hat{x}_s(t) = \hat{x}(t|t) + L\left[\hat{x}_s(t\bar{\gamma}) - \hat{x}(t\bar{\gamma}|t)\right] \tag{B.9}$$

We now proceed to show that (B.8) indeed holds and compute explicitly the matrix L. We begin with the following iterated expectation.

$$E[\tilde{x}(t|t)|\nu_{\bar{t}|t}] = E[E[\tilde{x}(t|t)|\tilde{x}(t\bar{\gamma}|t), \nu_{\bar{t}|t}]|\nu_{\bar{t}|t}] \tag{B.10}$$

We now examine the inner expectation, $E[\tilde{x}(t|t)|\tilde{x}(t\bar{\gamma}|t), \nu_{\bar{t}|t}]$, in detail. In particular note that $Y_{\bar{t}}$ corresponds to measurements at all nodes outside of the subtree with root note t – i.e. at all nodes including $t\bar{\gamma}$ and extending from it in either of the two directions other than toward t (i.e. toward $t\bar{\gamma}^2$ or toward $t\delta$–see Figure 1). Any such node is connected to $t\bar{\gamma}$ by a backward segment (of possibly zero length) moving back to $t\bar{\gamma}^r$ for some $r > 0$, followed by a forward segment moving down the other portion of the tree (i.e. a path of the form $t\bar{\gamma}, t\bar{\gamma}^2, \ldots, t\bar{\gamma}^{r-1}, t\bar{\gamma}^r, t\bar{\gamma}^{r-1}\delta, t\bar{\gamma}^{r-1}\delta\alpha, \ldots$). By using the backward dynamics (2.13)-(2.16) for the backward portion of any such path and the forward model (2.1) for the forward segment, we can express $x(s)$ for any s for any s outside the tree as a linear combination of $x(t\bar{\gamma})$, backward white noises $\tilde{W} = \{\tilde{w}(t\bar{\gamma}^r)|r > 0\}$ and forward white noises $W = \{w(s)|s\epsilon t\bar{\gamma}\bar{\gamma}^*\delta\{\alpha, \beta\}^*\}$. Then, letting V denote the vector of measurement noises for $Y_{\bar{t}}$, we have that $Y_{\bar{t}}$ has the following form

$$Y_{\bar{t}} = L_1 x(t\bar{\gamma}) + f(\tilde{W}, W) + V \tag{B.11}$$

where f is a linear function of its arguments. Note that by construction $\tilde{W}$ and W are orthogonal to $x(t\bar{\gamma})$ and to everything beneath $x(t\bar{\gamma})$. The same is trivially true of V. Thus, in particular $f(\tilde{W}, W) + V \perp Y_t$, and substituting (B.11) into (B.1) then yields

$$\nu_{\bar{t}|t} = L_1 \tilde{x}(t\bar{\gamma}|t) + f(\tilde{W}, W) + V \tag{B.12}$$

From this, together with the fact that $f(\tilde{W}, W) + V \perp \tilde{x}(t|t)$, $\tilde{x}(t\bar{\gamma}|t)$, yields

$$E[\tilde{x}(t|t)|\tilde{x}(t\bar{\gamma}|t), \nu_{\bar{t}|t}] = E[\tilde{x}(t|t)|\tilde{x}(t\bar{\gamma}|t)] \tag{B.13}$$

But by using (2.13) and (3.9) (in the latter of these with $t\alpha \mapsto t$ and $t \mapsto t\bar{\gamma}$) we find that

$$E[\tilde{x}(t|t)|\tilde{x}(t\bar{\gamma}|t)] \quad = \quad J(t)\tilde{x}(t\bar{\gamma}|t) \tag{B.14}$$

where $J(t)$ is given by (3.18). This, together with (B.10) then yields

$$E[\tilde{x}(t|t)|\nu_{\overline{t}\rceil t}] \;=\; J(t)E[\tilde{x}(t\overline{\gamma}|t)|\nu_{\overline{t}\rceil t}] \tag{B.15}$$

This together with (B.8) and (B.9) then yields $t\overline{\gamma}$ (3.17).

Finally, we can now easily derive a recursion for the smoothing error. Let

$$\tilde{x}_s(t) \triangleq x(t) - \hat{x}_s(t) \tag{B.16}$$

Subtracting $x(t)$ from both sides of (3.17) and rearranging terms we get

$$\tilde{x}_s(t) - J(t)\hat{x}_s(t\overline{\gamma}) = \tilde{x}(t|t) - J(t)\hat{x}(t\overline{\gamma}|t) \tag{B.17}$$

By multiplying both sides of (B.17) on the right by its transpose and taking expectations, we get

$$P_s(t) + J(t)E[\hat{x}_s(t\overline{\gamma})\hat{x}_s^T(t\overline{\gamma})]J^T(t) \;=\; P(t|t) + J(t)E[\hat{x}(t\overline{\gamma}|t)\hat{x}^T(t\overline{\gamma}|t)]J^T(t) \tag{B.18}$$

$$P_s(t) \;=\; E[[\tilde{x}_s(t)\tilde{x}_s^T(t)] \tag{B.19}$$

where we have relied on the fact that

$$E[\tilde{x}_s(t)\hat{x}_s^T(t\overline{\gamma})] \;=\; 0 \tag{B.20}$$

$$E[\tilde{x}(t|t)\hat{x}^T(t\overline{\gamma}|t)] \;=\; 0 \tag{B.21}$$

And finally, since

$$E[\hat{x}_s(t\overline{\gamma})\hat{x}_s^T(t\overline{\gamma})] \;=\; P_x(t\overline{\gamma}) - P_s(t\overline{\gamma}) \tag{B.22}$$

$$E[\hat{x}(t\overline{\gamma}|t)\hat{x}^T(t\overline{\gamma}|t)] \;=\; P_x(t\overline{\gamma}) - P(t\overline{\gamma}|t) \tag{B.23}$$

we obtain (3.19).

References

[1] B. Anderson and T. Kailath, "Forwards, backwards, and dynamically reversible Markovian models of second-order processes", *IEEE Trans. Circuits and Systems*,CAS–26, no. 11, 1978, pp. 956-965.

[2] M. Barnsley, *Fractals Everywhere*, Academic Press, San Diego, 1988.

[3] M. Basseville, A. Benveniste, and A.S. Willsky "Multiscale Autoregressive Processes, Part I: Schur-Levinson Parametrizations", *IEEE Transactions on Signal Processing,* to appear.

[4] M. Basseville, A. Benveniste, and A.S. Willsky "Multiscale Autoregressive Processes, Part II: Lattice Structures for Whitening and Modeling", *IEEE Transactions on Signal Processing*, to appear.

[5] M. Basseville, A. Benveniste, A.S. Willsky, and K.C. Chou, "Multiscale Statistical Processing: Stochastic Processes Indexed by Trees," in *Proc. of Int'l Symp. on Math. Theory of Networks and Systems*, Amsterdam, June 1989.

[6] A. Benveniste, R. Nikoukhah, and A.S. Willsky, "Multiscale System Theory", Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, December 1990.

[7] G. Beylkin, R. Coifman, V. Rokhlin, "Fast Wavelet Transforms and Numerical Algorithms I", to appear in it Comm. Pure and Appl. Math.

[8] A. Brandt, "Multi-level adaptive solutions to boundary value problems," Math. Comp. Vol. 13,1977, pp.333-390.

[9] W. Briggs, "*A Multigrid Tutorial*, SIAM, Philadelphia, PA, 1987.

[10] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Comm.*, vol. 31, pp. 482–540, 1983.

[11] K.C. Chou, S. Golden and A.S. Willsky, "Modeling and Estimation of Multiscale Stochastic Processes", *Int'l Conference on Acoustics, Speech, and Signal Processing"*, Toronto, April 1991.

[12] S.C. Clippingdale and R.G. Wilson, "Least Squares Image Estimations on a Multiresolution Pyramid", Proc. of the 1989 Int'l Conf. on Acoustics, Speech, and Signal Proceeding.

[13] R.R. Coifman, Y. Meyer, S. Quake and M.V. Wickehauser, "Signal Processing and Compression with Wave Packets", preprint, April 1990.

[14] I. Daubechies,"Orthonormal bases of compactly supported wavelets", Comm. on Pure and Applied Math. 91, 1988, pp. 909-996.

[15] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. on Information Theory*, 36, 1990, pp. 961-1005.

[16] P. Flandrin, "On the Spectrum of Fractional Brownian Motions", *IEEE Transactions on Information Theory*, Vol. 35, 1989, pp. 197-199.

[17] S. Golden, *Identifying Multiscale Statistical Models Using the Wavelet Transform*, S.M. Thesis, M.I.T. Dept. of EECS, May 1991.

[18] A. Grossman and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape", SIAM J. Math. Anal. 15, 1984, pp. 723-736.

[19] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, v. 17, pp. 185-203, 1981.

[20] S.G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Anal. and Mach. Intel.*, Vol. PAMI-11, July 1989, pp. 674-693.

[21] S.G. Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models", *IEEE Transactions on ASSP*, Vol. 37, December 1989, pp. 2091-2110.

[22] B. Mandelbrot, *The Fractal Geometry of Nature,* Freeman, New York, 1982.

[23] B. B. Mandelbrot and H.W. Van Ness, "Fractional Brownian Motions, Fractional Noises and Applications", *SIAM Review*, Vol. 10, October 1968, pp. 422-436

[24] Y. Meyer, "L'analyse par ondelettes", Pour la Science, Sept. 1987.

[25] A. P. Pentland, "Fractal-Based Description of Natural Scenes", *IEEE Transactions on Patt. Anal. and Mach. Intel.,* Vol. PAMI-6, November 1989, 661-674.

[26] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum Likelihood Estimates of Linear Dynamic Systems," AIAA Journal, Vol. 3, No. 8, Aug. 1965, pp. 1445-1450.

[27] A. Rougee, B. Levy, and A. Willsky, "An estimation-based approach to the reconstruction of optical flow," Laboratory for Information and Decision Systems Technical Report, no. LIDS-P-1663, MIT, April 1987. 1445-1450.

[28] M.J. Smith and T.P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders", IEEE Trans. on ASSP 34, 1986, pp. 434-441.

[29] G. Strang, "Wavelets and Dilation Equations: A Brief Introduction", *SIAM Review*, Vol. 31, No. 4, December 1989, pp. 614-627.

[30] R. Szeliski, "Fast Surface Interpolation Using Hierarchical Basis Function", *IEEE Transactions on PAMI*, Vol. 12, No. 6, June 1990, pp. 513-528.

[31] D. Terzopoulos, "Image Analysis Using Multigrid Relaxation Methods", *IEEE Transaction on PAMI*, Vol. PAMI-8, No. 2, March 1986, pp. 129-139.

[32] A.H. Tewfik and M. Kim, "Correlation Structure of the Discrete Wavelet Coefficients of Fractional Brownian Motions", submitted to *IEEE Transactions on Information Theory.*

[33] M. Todd and R. Wilson, "An Anisotropic Multi-Resolution Image Data Compression Algorithm", Proc. of the 1989 Int'l Conf. on Acoustics, Speech, and Signal Processing.

[34] G.C. Verghese and T. Kailath, "A further note on backward Markovian models," *IEEE Trans. on Information Theory*, IT–25, pp. 121–124, 1979.

[35] M. Vetterli, and C. Herley, "Wavelet and Filter Banks: Relationships and New Results", *Proceedings of the ICASSP*, Albuquerque, NM, 1990.

[36] G.W. Wornell, "A Karhunen-Loeve-Like Expansion for 1/f Processes via Wavelets", *IEEE Transactions on Information Theory*, Vol. 36, No. 9, July 1990, pp. 859-861.

[37] G.W. Wornell and A.V. Oppenheim, "Estimation of Fractal Signs from Noisy Measurements Using Wavelets", *IEEE Transactions on Signal Processing,* to appear.

[38] A. Witkin, D. Terzopoulos and M. Kass, "Signal Matching Through Scale Space", *Int. J. Comp. Vision*, Vol 1, 1987, pp. 133-144.

[39] R.B. Washburn and A.S. Willsky, "Optional Sampling of Submartingales on Partially-Ordered Sets", *Annals of Prob.* , Vol. 9, No. 6, 1981, pp. 957-970.

[40] T.M. Chin, "Dynamic Estimation in Computational Vision", Ph.D. Thesis, M.I.T. Dept. of EECS, Oct. 1991.

[41] R.E. Crochiere and L.R. Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ 1983.

[42] K.C. Chou, "A Stochastic Modeling Approach to Multiscale Signal Processing," M.I.T. Department of Electrical Engineering and Computer Science, Ph.D. Thesis, May 1991.

[43] K.C. Chou, A.S. Willsky and R. Nikoukhah, "Multiscale Systems, Kalman Filters, and Riccati Equations", to be submitted for publication.