Large-Scale 2D Dynamic Estimation

by

Fakhry Mahmoud Khellah

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2001

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Dynamic estimation, the assimilation of data over time, is an important scientific issue in remote sensing, image processing, and computer vision, to name a few.

The main motivation for this thesis is large-scale 2-D dynamic estimation problems related to remote sensing. For such problems, number of variables to be estimated can reach to the order of millions. As a result, direct application of conventional estimation algorithm, i.e., the Kalman filter, becomes totally impractical from two technical aspects: computational and storage demands. In this thesis, we propose a new method for large-scale 2-D estimation problems that emulates the Kalman filter, but with more efficient computational and storage demands.

Using *parameterized* error models to model the huge error covariance matrices is the main contribution of this thesis. Under this scope, we develope a new approximate error prediction step and a new approximate large-scale update step.

We studied the performance of the proposed method in the context of small synthetic 2-D diffusion processes. In addition, we applied our method to a large-scale remote sensing problem: the estimation of the ocean surface temperature based on sparse satellite measurements.

Acknowledgements

First, I would like to express my deepest gratitude and thanks to my thesis supervisor Prof. Paul Fieguth. I shall admit that without his guidance, encouragement, and great ideas this effort would not have been possible. I appreciate the considerable amount of time and effort Dr. Paul gave to this thesis. I am grateful for the huge amount of knowledge I gained through him. I am thankful for his continuous helpful advices during my course of study.

I would like to thank Prof. Ed Jernigan for his guidance and input to this research.

I would like to thank Prof. Dan Stashuk for carefully reviewing and proofreading my thesis. I would like also to thank all my committee members for their comments on this work.

I would like to thank my brother Dr. Mohammed Khellah and his family for their support during the past four years.

I would like to thank all my colleagues in the VIP research group for their support, and useful discussions. In particular, I would like to thank John Morton for taking care of the VIP lab. computer systems.

I would like also to thank all my friends and in particular Salah Kanoun for introducing me to Dr. Paul.

I am grateful for my parents *Nahidah and Mahmould* for their continuous support and encouragement.

Lastly, I am thankful for the National Science and Engineering Research Council (NSERC), and Ontario Graduate Scholarship (OGS) program for the financial support.

Contents

List of Tables

List of Figures

Notation

l	Spatial correlation length
k	Total number of intermediate interpolating priors
l_i	i^{th} correlation length used to construct interpolant prior $P(l_i)$, $1 \leq i \leq k$.
$P(l)$	2D isotropic stationary prior with correlation length l
$\tilde{P}(l)$	Exact estimation error covariance based on prior $P(l)$
$\zeta(l)$	Sample path from prior $P(l)$
$\widehat{\underline{x}}(l)$	Exact estimates for state $\underline{x}$ based on prior $P(l)$.
$\tilde{\mathbf{p}}(l)$	Exact estimation error variances based on prior $P(l)$.
$\widehat{\underline{x}}$	Approximate estimates for the whole 2D field computed based on interpolating weights.
$\widehat{\tilde{\mathbf{p}}}$	Approximate error variances for for the whole 2D field computed based on interpolating weights.
$\underline{\alpha_x}(l)$	State interpolating weight associated with correlation length l
$\underline{\alpha_p}(l)$	Estimation error interpolating weight for correlation length l
N	The length in pixels of one edge of the 2D domain
n	Number of elements in field, 2D $n = N \times N$
$\Gamma(l)$	A region of the 2D domain that has correlation length equal to l

Chapter 1

Introduction

1.1 Thesis Motivations

Dynamic estimation, the assimilation of data over time, is an important scientific issue in remote sensing, image processing, and computer vision, to name a few.

Large scale 2-D dynamic estimation problems related to the remote sensing is the main motivation for this thesis. For such problems, the number of variables to be estimated can reach to the order of millions. As a result, direct application of conventional estimation algorithms, i.e., like the Kalman filter, becomes totally impractical from two technical aspects: computational and storage demands.

From the computational aspect, the Kalman filter (to be detailed in the next chapter) involves two highly computationally demanding operations: matrix multiplication and matrix inversion. Both have computational cost of order $\mathcal{O}(N^3)$ where N is the length of the state vector. For a 2D process of size $N \times N$ the computational cost is $\mathcal{O}(N^6)$. The huge storage demands are essential since the filter requires the full error covariances, each of size $N^2 \times N^2$, to be readily available at each time step.

From the above, it becomes obvious that for large-scale 2-D dynamic estimation problems there exists a need to develop an *approximate* dynamic filter that emulates the

Kalman filter but with more efficient computational and storage demands.

Although there have been several research works that tried to deal with such issues for *relatively* large estimation problems, these studies mainly addressed the computational demands of the *update* step under some restricted and impractical assumptions:

- Fast Fourier transform (FFT) [33] methods require that the underlying process is spatially stationary.
- Iterative methods for solving the normal equations [48] require the full error matrices to be available. In addition they do not explicitly provide the estimation error statistics.
- The sparse Kalman filter [4, 15, 16] exploits the sparsity of the filter matrices and uses polynomial approximation for matrix inversion. This method requires the availability of the error matrices. In addition it requires that the matrix to be inverted be diagonally dominant.
- Covariance extension methods [20, 40, 50] reduce the storage requirements for the error covariance matrices but are only developed for 1-D large estimation problems.
- Multiscale based dynamic estimation for 2D problems [50] with reduced order tree states is only applicable for small size problems.

Another motivation for this thesis is to extend the capability of a recently developed efficient static estimator, the multiscale estimator [17, 32, 50, 53, 67], to include large-scale 2-D dynamic estimation problems. The multiscale estimator is a static estimator that can efficiently solve large-scale static estimation problems. It provides estimates in addition to estimation error variances and a model for the estimation error covariance at a very low computational and storage demands.

A third motivation for this research is to develop a dynamic estimation tool that can be applied to diffusive-like processes such as the ocean surface temperature (OST). Studying diffusive-like processes has many practical and useful applications in science and engineering. Since the ocean surface dynamics are not mainly diffusive, we want to point out that the (OST) problem is strictly considered as a demonstrative problem for this work.

1.2 Thesis Contributions

In this work, the main contribution is based on developing a dynamic estimator that deals with *parameterized* error models instead of very large error covariance matrices. We investigate this approach for 2-D *diffusion* processes. One of the challenges that we are trying to address is that while the process covariance structure is stationary, the error process covariance is not. Theoretically, there is no way to have a direct formulation for non-stationary parametric error model that can be fitted to a non-stationary process statistics. Our approach to this is addressed in the context of the following two steps.

1.2.1 Approximate error prediction models

In this context, we investigated a new error prediction approach that suits large-scale 2-D dynamic estimation problems for 2-D diffusion processes. This approach is simply based on exploiting the fact that any covariance matrix can be written in terms of its diagonal elements and a correlation coefficient matrix. We exactly propagate the error variances and model the correlation matrix by a parametric model. By establishing an empirical model that relates the estimation error variances to the correlation length at any element of the 2-D error process we were able to encode the non-stationarity in the predicted and updated error models.

1.2.2 Non-stationary large-scale 2-D update step

The measurement update step in the Kalman filter is a static estimation problem. The multiscale estimator can efficiently be used to solve the update step for large-scale 2-D dynamic estimation problems. Estimates computed by the multiscale estimator are in general based on *stationary* prior. In the dynamic estimation context, the correlation structure of the error process is generally non-stationary. One of the contributions of this work is to be able to compute non-stationary estimates consistent with the correlation structure of the error process based on stationary priors.

1.2.3 Application to OST

As mentioned earlier, the main motivation of this work is to be able to apply it to solve large-scale 2-D dynamic estimation problems related to diffusion processes. The problem that we used to demonstrate our method is the estimation of the ocean surface temperature (OST) based on satellite observations. The size of the images is huge (512×512) elements. The problem is hard from several aspects:

- The data are very sparse;
- The size of the state vector is very large;
- Dynamic estimation requires the availability of a dynamic model.

The first two issues motivate the development of our approximate dynamic estimation method. The issue of the temporal dynamics for the ocean surface is solved by assuming that the ocean surface is diffusive. We want here to emphasize the following issues:

The ocean surface dynamics are not exactly diffusive. However, the diffusion assumption for the ocean surface temperature is reasonable because it covers the current mixing effect and the heat transfer to the surroundings. An advective component in the ocean dynamics which is due to the effect of currents is not incorporated by considering

only the diffusion model. Ideally, obtaining exact ocean diffusion dynamics parameters requires collaboration with oceangraphers. Our effort is concentrated on developing a dynamic estimation tool for large-scale problems that can be of potential interest to oceangraphers. The adopted approach for obtaining approximate diffusion parameters for the ocean surface is based on fitting a diffusion process statistics to the ocean statistics inferred from the data.

Despite the problem size, and the lack of true ocean surface dynamic model we were able to successfully demonstrate the results of the proposed approach. Quantitative scientific assessment of the results would require the input of oceangraphers.

1.3 Thesis Organization

This thesis divides into seven chapters. This chapter has presented a summary on both the motivations and the contributions of this work.

Chapter 2 presents general background material on estimation . The material is meant to be concise yet comprehensive for readers who are not familiar with estimation theory. The chapter addresses the two main estimation categories: static estimation and dynamic estimation. Detailed background of a standard dynamic estimation algorithm (the Kalman filter) is presented. In addition, alternative approaches when dealing with large-scale dynamic estimation problems are reviewed. A detailed discussion on various approaches to test the performance of the approximate filters is mentioned in the last section.

Chapter 3 presents the background material on multiscale estimation. This chapter covers the general concept of the multiscale estimation. We revisited some of the previous development and applications of this framework. In addition, we review the limitations of the previous work on large-scale dynamic estimation using the multiscale estimator.

The contributions of this research are presented in chapters 4 through 6. In Chapter 4,

we present our approach to approximate error predictions in the context of diffusion dynamics. We show how we can use stationary parameterized error models to propagate the updated estimation error variances. By empirically studying the relationship between the error variances and the correlation length we are able to encode the non-stationarity in the error process statistics.

Chapter 5 presents our method for computing estimates based on parameterized error models but taking into consideration the spatial non-stationarity in the predicted error model. Our method makes use of the efficiency of the multiscale estimator in computing stationary estimates. By properly combining stationary estimates computed based on several stationary priors we are able to produce approximate non-stationary estimates and their associated estimation error variances. We investigate the effect of the number of stationary priors on the estimation quality. In the last section, we show some synthetic static estimation experiments to illustrate our approach.

Chapter 6 presents several synthetic examples on dynamic estimation for 2D diffusion processes in addition to the ocean surface temperature problem. We compare the results obtained by our method to the estimates produced by the standard Kalman filter algorithm. In the last section, we present the ocean surface temperature problem and we show that our method is capable of producing approximate estimates if reasonable ocean surface statistics and diffusion parameters are available.

Finally, Chapter 7 presents a summary of this work and topics for future work.

Chapter 2

General Background

This chapter presents general background related to the subject of estimation. The material is meant to be concise for readers who are not familiar with estimation theory. The chapter addresses the two main estimation categories: static estimation and dynamic estimation. The reader may think that the two problems are unrelated, but infact, they are closely related and are presented in this chapter.

Section 2.1 talks about the estimation problem in general. Static estimation is presented in Section 2.2. First, the two main types of static estimation are presented. Next, some important properties for estimating Gaussian variables are listed. For large-scale static estimation problems, there exist certain techniques that can deal with the computational and storage issues. Properties and drawbacks of these methods are then addressed.

Dynamic estimation background, which is closely related to the topic of this thesis, is presented in Section 2.3. This section starts by presenting the Kalman filter basis. Then, two important alternative forms of the standard Kalman filter are detailed. Next, the filter limitations with respect to computational and storage issues are addressed. In addition, various approaches to dealing with large-scale dynamic estimation problems are presented. Finally, Section 2.4 discusses various approaches to test the performance of the approximate filters.

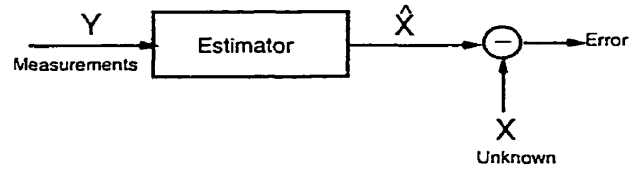


Figure 2.1: Illustration of the general estimation problem. y is a set of observations for unknown quantity x. $\hat{x}$ is an estimate of x computed by an estimator derived based on minimizing some function of the the estimation error $x - \hat{x}$.

2.1 Overview

In many scientific or engineering problems, scientists are interested in obtaining a mathematical model describing the underlying process that can be used in simulating and analyzing the underlying phenomena. Normally, the construction of such models requires the determination of unknown parameters, typically based on experimentally-measured quantities. However, in collecting such measurements, two issues arise:

- Not all quantities can be measured;
- Most sensors or measuring devices introduce errors into the measurement process. For those measurable quantities, how can one filter out the sensor noise?

Estimation theory [41, 65, 71, 91] plays a major role in dealing with the above two issues. Estimation generally refers to the subject of making inferences about some unknown (random) quantities based on measurements of the same or some related unknown (random) variables. Estimation problems are mainly composed of five major components [60, 69, 92]:

1. The variables to be estimated. Based on the underlying problem, the quantity to be estimated can either be a scalar (a single value) x or a vector $\underline{x} \in R^n$. The vector quantity can represent a one-dimensional process or a lexicographically stacked

two-dimensional one. Estimation problems are categorized into two main types depending on the nature of the quantity to be estimated. If the quantity to be estimated is just a deterministic unknown, then this is called parameter estimation or non-Bayesian estimation [65, 84]. On the other hand, if the unknown quantity is random, then this is called Bayesian estimation [65, 84].

2. A set of measurements or observations y.

3. A mathematical model describing the relationship between the observations y and the unknown quantity x in addition to the nature of the measurement noise. The measurement model is generally a linear function of the unknown. That is, the obtained measurements are linear combinations of the unknown quantities. In addition, a probabilistic or mathematical model describing the uncertainties in the measurements is usually known. In most cases, the model is given by the measurement noise covariance matrix R which is usually diagonal (multiple of the identity), i.e., the noise components are independent.

4. A probabilistic or mathematical model describing the prior knowledge of the underlying random process. This is an essential component only in the case of estimating a random process. Usually the model is given in terms of the process covariance matrix P_x. In the case of a deterministic unknown x, then a prior knowledge for the distribution of the measurements is required. This is usually given in terms of the measurements probability density function (PDF).

5. A performance criterion upon which the estimator is derived and its performance is assessed.

Figure 2.1 summarizes the general estimation problem. To clarify the above points the following example is presented:

Given a set of observations y for a random quantity x, the relationship between y and x is given by

$$y = Cx + v \tag{2.1}$$

Equation (2.1) is a common linear mathematical model for the relationship between measurements and unknown quantities. The term v is an additive Gaussian white noise with mean 0 and covariance R. These statistics represent a probabilistic model for the measurement noise. The unknown quantity x is random and it is assumed to be Gaussian. This statistical model is called a prior. In this case, the prior for x can be described by its mean μ_x and covariance matrix P_x. Given the above quantities, in order to estimate x an estimation criterion must be specified. A common criterion is minimizing the mean square estimation error $E[(x - \hat{x})^2]$. Now, the estimation problem is completely identified. The derivation for the estimate and its error covariance is given in the following section.

2.2 Static estimation

In static estimation the unknown quantity does not evolve with time. In addition, the sampled measurements are not obtained temporally. So static estimation deals with estimating a static quantity based on a single set of measurements [60, 65].

As mentioned in Section 2.1, estimation problems can be categorized as Bayesian or non-Bayesian, depending on the statistical nature of the unknown quantity to be estimated. In the following two subsections, each of these two cases is discussed in turn.

2.2.1 Non-Bayesian estimation

When the unknown quantity x is deterministic as opposed to random the estimation problem becomes non-Bayesian. Estimating the mean or the variance of a distribution from

a set of observations are examples of non-Bayesian estimation.

The most common criterion used in non-Bayesian estimation is the maximum likelihood [60]. Let $\underline{y}$ be the measurement vector and $\underline{x}$ is an unknown quantity, then the estimate for $\underline{x}$ is the value of $\underline{x}$ which maximizes the likelihood function $p(\underline{y}|\underline{x})$.

Consider the basic static estimation problem [41, 60, 65] in which the goal is to estimate a collection of *unknown* parameters represented by a vector $\underline{x} \in \mathcal{R}^n$, based on a set of noisy measurements represented by $\underline{y} \in \mathcal{R}^m$. Generally, we have a linear relationship between the unknown $\underline{x}$ and the measurements $\underline{y}$ given by

$$\underline{y} = C\underline{x} + \underline{v} \tag{2.2}$$

Where the matrix C describes the nature of the measurement process. Generally, C is a "selection matrix" indicating the measured components and which combinations of x_i corresponds to the measurement y_i. If C is a selection matrix then it has ones in the columns that correspond to the measured quantity. In the following, we will refer to (2.2) as the *linear model*. Observe that $p(\underline{Y}|\underline{X})$ is also Gaussian with mean $E[\underline{y}] = C\underline{x}$ and covariance R.

$$p(\underline{Y}|\underline{X}) = \frac{1}{(2\pi)^{\frac{n}{2}}|R|^{1/2}} \exp\left[-\frac{1}{2}(\underline{y} - C\underline{x})^T R^{-1}(\underline{y} - C\underline{x})\right] \tag{2.3}$$

In order to maximize (2.3) we need to minimize

$$J = \frac{1}{2}(\underline{y} - C\underline{x})^T R^{-1}(\underline{y} - C\underline{x}) \tag{2.4}$$

Note that using the inverse of the measurement noise covariance R^{-1} to weight the sum of squares of the error is a special case of the weighted least squares where we seek an estimate $\hat{x}$ that will minimize the weighted sum of squares of the error by some arbitrary positive definite weighting matrix W [65]. By differentiating (2.4) with respect to $\underline{x}$ and

equating the result to zero, the optimal estimate for the linear model is given by [65]

$$\hat{\underline{x}} = (C^T R^{-1} C)^{-1} C^T R^{-1} \underline{y} \tag{2.5}$$

The corresponding estimation error covariance is [65]

$$E[(\underline{x} - \hat{\underline{x}})(\underline{x} - \hat{\underline{x}})^T] = \tilde{P} = (C^T R^{-1} C)^{-1} \tag{2.6}$$

2.2.2 Bayesian estimation

The second type of estimation problem is when the unknown quantity is a random vector $\underline{x}$. In this case, we have prior knowledge about $\underline{x}$ specified by its density function $p(\underline{x})$. Bayesian estimation provides a method to merge the external knowledge (i.e., the prior $p(\underline{x})$) and the observations $\underline{y}$ in estimating $\underline{x}$.

There are several sources for prior knowledge. The prior knowledge can be obtained from data previously studied by estimating the probability density function of the unknown quantity, or estimating the joint PDF of the measurements and the unknown $p(\underline{x}, \underline{y})$ [8], or it can be an imposed constraint, like a smoothness constraint [33].

In the general Bayesian framework, two main components affect the estimation results: the prior knowledge of both the measurements and the unknown and the chosen estimation cost function $c(\underline{e})$. The estimate $\hat{\underline{x}}(\underline{y})$ is chosen to minimize the expected value of the estimation cost function (i.e., the average cost)

$$E[c(\underline{e})] = E[C(\underline{x} - \hat{\underline{x}})] \tag{2.7}$$

There are various estimation criteria based on the selection of the cost function [28]

- Mean-square estimation, which selects an estimate $\hat{\underline{x}}$ that minimizes the mean

square error

$$c(\underline{e}) = \| (\underline{x} - \hat{\underline{x}}) \|^2$$

This is referred to as Bayesian *Least squares estimation* LSE [28, 65, 84].

- Maximum posterior estimation MAP [28, 65, 84] which uses a uniform cost function given by

$$c(\underline{e}) = \begin{cases} 1 & |\underline{x} - \hat{\underline{x}}| \geq \epsilon \\ 0 & |\underline{x} - \hat{\underline{x}}| < \epsilon \end{cases}$$

  Observe that the MAP criterion treats all the errors above a given value ϵ equally by a cost function that is equal to one. The above cost criterion corresponds to selecting an estimate $\hat{\underline{x}}$ that maximizes the posterior probability $p(\underline{x}|\underline{y})$

- Linear least squares estimation LLSE criterion is an extension to the LSE. However, in LLSE we require the estimate to be a linear function of the measurements $\hat{\underline{x}} = A\underline{y} + \underline{b}$. The estimation cost in this case is given by

$$c(\underline{e}) = \| (\underline{x} - \hat{\underline{x}}) \|^2$$

The Bayes' least squares estimate LSE is the conditional mean of $\underline{x}$ [28, 65, 84].

$$\hat{\underline{x}}_B(\underline{y}) = E[\underline{x}|\underline{y}] \tag{2.8}$$

In most cases, the conditional mean is not a linear function of the observations. LLSE criterion assures that the estimate is a linear function of the observations. For the Gaussian case, the conditional mean given by (2.8) is a linear function of the measurements. Therefore, estimates produced by LSE and LLSE are equivalent for the Gaussian case [2, 65, 71].

Following is the derivation for the estimate for the Gaussian case [2, 65, 71] where the measurement *linear model* given in (2.2) is considered. Note that the derivation is based on LSE criterion which is equivalent to LLSE in this case.

The problem statistics are

$$E[\underline{v}] = 0 \quad E[\underline{v}\underline{x}^T] = 0 \quad E[\underline{v}\underline{v}^T] = R \quad E[\underline{x}] = \underline{m}_x \quad E[\underline{x}\underline{x}^T] = P_x \tag{2.9}$$

This makes the estimation error covariance equal to the sum of both the process covariance and the measurement noise covariance

$$P_y = CP_xC^T + R \tag{2.10}$$

The least-squares criterion in this case becomes finding $\underline{\hat{x}}$ that will minimize the mean-square error given by

$$E[(\underline{x} - \underline{\hat{x}})^T(\underline{x} - \underline{\hat{x}})] \tag{2.11}$$

In the case where all the random variables are Gaussian, the estimate given by (2.8) is just the conditional mean of the conditional distribution $p_{x|y}(X|Y)$which is also Gaussian. The estimate in this case is given by

$$\underline{\hat{x}} = \underline{m}_x + P_xC^T(CP_xC^T + R)^{-1}(\underline{y} - C\underline{m}_x) \tag{2.12}$$

and the associated estimation error covariance is

$$E[\underline{\tilde{x}}\underline{\tilde{x}}^T] = \tilde{P} = P_x - P_xC^T(CP_xC^T + R)^{-1}CP_x \tag{2.13}$$

where $\underline{\tilde{x}} = \underline{e} = \underline{x} - \underline{\hat{x}}$ is the estimation error.

It is important to mention the alternative forms of equations (2.12), (2.13) which are

derived using the ABCD lemma [28]

$$\hat{\underline{x}} = \underline{m_x} + \tilde{P}C^T R^{-1}(\underline{y} - C\underline{m_x}) \quad (2.14)$$

$$\tilde{P} = (P_x^{-1} + C^T R^{-1} C)^{-1} \quad (2.15)$$

This form is identical to the one given in (2.12),(2.13) with respect to the estimation results. However, the computational cost of both forms can be different, as will be illustrated in Section 2.2.4.

2.2.3 Estimation properties of the Gaussian case

A Gaussianity assumption appears in many estimation problems for at least two important reasons. First, Gaussian random variables, in many natural phenomena, provide a reasonable and simple approximation. Second, if the underlying random phenomenon is the superposition of an arbitrary large number of random processes, the whole process can be considered as Gaussian.

There are several important properties of estimates when the involved random variables are Gaussian [71, 84]:

- Estimates can be derived based on the second-order statistics only instead of the whole joint probability density function.
- The LSE estimate, which is the conditional mean, is always a linear function of the observations, making the LSE criterion and LLSE criterion equivalent.
- The estimation error covariance is independent of the observations. This is an important feature that allows assessing the estimates quality before setting up the experiments and taking the measurements.
- Estimates are optimal (i.e., unbiased and efficient [71]) with respect to any of the previously mentioned criteria.

2.2.4 Approaches for Large Scale static Estimation

Estimation problems related to remote-sensing involve dealing with huge amount of information. Global ocean modeling and climate studies, based on satellite altimetry measurements, are two important examples. In large scale problems, the number of measurements and the number of variables to be estimated can reach the order of millions. In this case, the LLSE suffers from high computational cost and the impossibility of storing a million by million covariance matrix.

Consider applying the LLSE to a 2-D random field of size $n = N \times N$. This involves the inversion of a matrix of size $N^2 \times N^2$, requiring $\mathcal{O}(N^6)$ computations (matrix inversion is of order $\mathcal{O}(n^3)$). In addition, the storage requirements are of order $\mathcal{O}(n^2)$.

The difficulty in applying the LLSE to large-scale problems is associated with the matrix inversion and multiplication required to compute both the estimate and the error covariance.

The *first* intuitive approach when dealing with large-scale static estimation problems is to determine which form of the LLSE given in (2.12),(2.13), and (2.14),(2.15) should be employed. The selection should be based on the dimension of the measurements and the variables to be estimated. Both forms involve matrix inversion which is of high computational cost. However, in the first form, the size of the matrix to be inverted is related to the number of measurements. In the second form, the number of variables to be estimated determines the size of the matrix that needs to be inverted. If there is a considerable difference between the number of measurements taken and the number of variables to be estimated then a great improvement can be achieved by choosing the proper form. The *second* intuitive approach is to minimize the number of variables to be estimated. This can be done by ignoring those variables that are exactly measured. In some cases, this can lead to a considerable reduction in computational demands. However, this is not usually the case when applying the LLSE to large scale problems.

Because the solution of large two-dimensional least squares estimation problems is

of considerable interest in many disciplines, a number of efficient least squares methods have been proposed. Most of these methods address the matrix inversion problem. However, for large-scale problems matrix multiplication is still a problem to be considered. Following is a brief overview of some of these methods:

1: **Brute-force solution by a more efficient matrix inversion method**

Inversion of large matrices is impractical. However, in many problems, the matrix to be inverted has a well-defined structure (e.g. banded). In such cases, the inversion can be done efficiently by approximating it with infinite series [16]. Provided that the matrix to be inverted A is diagonally dominant, A can be written as $A = D+O$, where D is the main diagonal of A and O are the remaining off-diagonals. Then A^{-1} can be approximated by:

$$A^{-1} = D^{-1} - D^{-1}OD^{-1} + D^{-1}OD^{-1}OD^{-1} - D^{-1}OD^{-1}OD^{-1}OD^{-1} + \cdots \quad (2.16)$$

2: **Direct and Iterative methods** [43, 44, 96]

The problem of LLSE can be viewed as solving a system of linear equations i.e., $A\underline{\hat{x}} = b$. For example, consider writing equation (2.14) in the following form:

$$(P^{-1} + C^T R^{-1} C)\underline{\hat{x}} \;=\; C^T R^{-1} \underline{y} \quad (2.17)$$

Equation (2.17) is commonly known as the system of normal equations which are expressed as a linear system (i.e., $\underline{y} = A\underline{x}$).

Linear systems can be solved numerically by mainly two classes of methods: direct methods and iterative methods.

Direct methods give the exact solution in a finite number of elementary arithmetic operations provided there are no rounding errors. There are three categories of the direct methods [44, 94]:

1. Gaussian elimination with interchanges
2. Triangular factorization including Cholesky factorization for positive-definite matrices
3. Householder reduction to upper triangular form.

Direct methods can be impractical if the coefficient matrix $[A|y]$ is very large and sparse, because the factorization can lead to dense factors.

Alternatively, *iterative methods* [42, 43, 44, 48] are used. A typical iterative method involves the initial selection of an approximation $x^{(1)}$ to x, and the determination of a sequence $x^{(2)}, x^{(3)}, \cdots$ such that $\lim_{i\to\infty} x^{(i)} = \hat{x}$. In practice, the iteration is stopped when the current approximation is acceptably close to $\hat{x}$. Several iterative methods are available: the Jacobi, Gauss-Seidel, Successive Over-Relaxation, and Conjugate Gradient methods.

Iterative methods suffer from the following problems:

- The Convergence rate in many cases is slow which leads to large number of iterations. Even methods which accelerate the convergence rate remain impractical for problems that have considerable size. The above mentioned iterative methods tend to reduce the high frequency components of the error rapidly but reduce the low frequency (i.e., smooth) components of the error much more slowly. This leads to the poor convergence rate.
- It is generally hard to determine the stopping criterion for the iterative method.

In addition to the above classical iterative methods, there exist other efficient iterative techniques such as the multigrid method [48, 96] and nested dissection methods [10, 51].

The multigrid method is usually used to solve discretized partial differential equations. It makes use of the idea that the high and low frequency components of the error

are relative to the mesh on which the solution is defined. A component that appears smooth on a fine grid may appear as high frequency on a coarser grid. Consequently, if we apply one of the iterative methods such as Gauss-Seidel on the coarser grid, then we may get rapid progress in removing the high frequency components of the error with very few iterations. Then, the results are interpolated back to the fine scale. On this scale, one might also apply Gauss-Seidel to ensure reducing the high frequency components. The final result is an approximate solution to the fine grid. Multigrid methods can converge in order $\mathcal{O}(n)$ operations where n is the number of grid points.

In most estimation problems, error statistics are as important as estimating the random variables. In dynamic estimation, error statistics for any time step are necessary for moving into the next time step. However, solving (2.17) using direct or iterative methods does not explicitly give estimation error statistics The above methods can provide error statistics by brute-force only. In fact, computing the estimation error covariance ($\tilde{P}$), generally, requires n times as much the computational work as finding the estimate $\hat{\underline{x}}$ where n is the number of the variables to be estimated.

3: **FFT Technique** [33]

In some cases, the underlying estimation problem is statistically toroidally stationary. For a 2-D process, toroidal stationarity is formally given by

$$E[x(i,j)x(i+\Delta x, j+\Delta y)] = E[x(0,0)x(\Delta x \mathrm{mod} M, \Delta y \mathrm{mod} N)] \qquad (2.18)$$

Where $\Delta x, \Delta y$ are the horizontal and vertical lags, respectively, and M, N are the 2-D field dimensions.

The basis of FFT methods is that the FFT diagonalizes any circulant matrix (i.e., each column/row can be obtained from the previous one by shifting all elements one place down/right and putting the last element at the top/left). Consequently, for stationary processes, the FFT can be used to diagonalize the matrices involved in the LLSE esti-

mation solutions (2.12),(2.13) and this will lead to the inversion of diagonal matrices which is of a very low computational cost. The FFT method, on the other hand, is limited by the two constraints that must be satisfied in the underlying estimation problem:

- it requires the measurements to be regular (i.e., we should have point measurements).
- it is only applied to stationary processes.

4: **Local methods** [37]

These methods compute local estimates based on measurements that are local to small regions obtained by dividing the whole large field into smaller regions. The main assumption in these methods is that neighbouring pixels are highly correlated where pixels that are a far distance from each other are only weakly correlated. This approach reduces the computational and storage demands required to solve the large problem, however, the estimates obtained in any region are only based on the local existing measurements. If a region is not measured then it will not be affected by neighbouring measurements. In addition, by only considering local correlation, high correlation information in the prior is ignored.

5: **Multiscale Method** [17, 37, 67]

This is an efficient, state of the art static estimation algorithm that can deal with large-scale static estimation problems. This method provides not only estimates but the associated estimation error variances. This algorithm is employed in this work. A complete background of the multiscale algorithm is presented in the next chapter.

2.3 Dynamic Estimation

In dynamic estimation problems, measurements evolve with time. In addition, the quantity to be estimated also evolves with time according to some dynamic rule. The

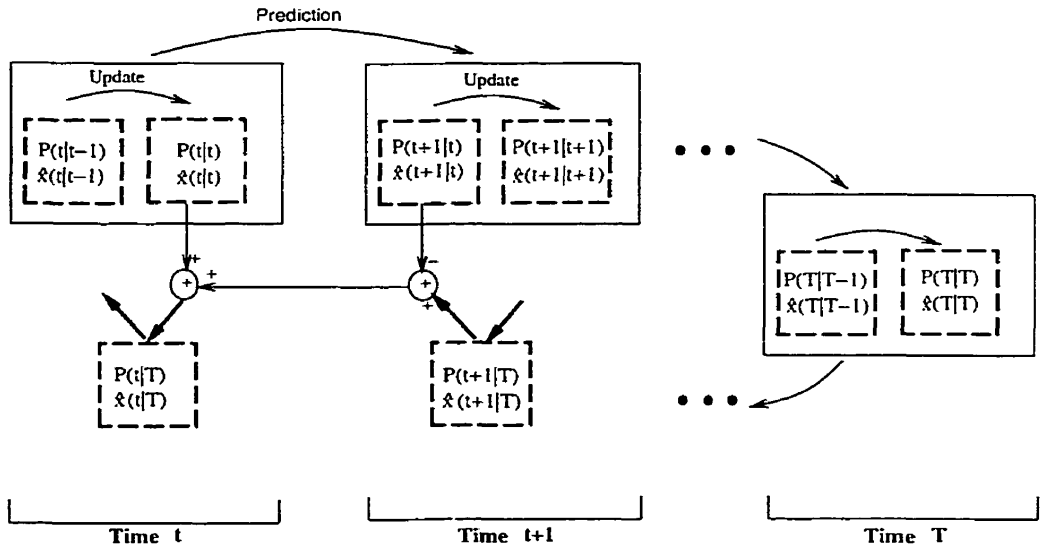


Figure 2.2: Forward and Backward smoothing.

system dynamics are usually governed by a partial differential equation. The following sections elaborate on the dynamic estimation subject.

2.3.1 Kalman filter formulation

The Kalman filter [2, 65, 71] is the conventional estimation tool that computes *optimal* state estimates and the associated estimation error covariance for an underlying dynamic system based upon a dynamic model and a dynamic measurement model. The Kalman filter is known to be the optimal estimator if all the involved filter quantities are Gaussian. In case of random quantities that have arbitrary statistics, the filter is the best *linear* estimator [65].

The linear time-invariant dynamic model for a discrete time process is given by the first-order Gauss-Markov model

$$\underline{x}(t+1) = A\underline{x}(t) + B\underline{w}(t) \quad \underline{x}(0) \sim \mathcal{N}(\underline{m}_0, P_0) \quad \underline{w}(t) \sim \mathcal{N}(0, I) \tag{2.19}$$

where,

$$\underline{m_0} = \hat{\underline{x}}(0|-1) = E[\underline{x_0}] \tag{2.20}$$

$$P_0 = P(0|-1) = E[(\underline{x_0} - E[\underline{x_0}])(\underline{x_0} - E[\underline{x_0}])^T] \tag{2.21}$$

where m_0 is the process mean and P_0 is an $n \times n$ (n is the dimension of the state vector) initial covariance matrix.

$\underline{x}(t)$: $(n \times 1)$ process state vector. Each element of $\underline{x}$ is a random variable that needs to be estimated

A : A matrix of size $(n \times n)$ which relates the state at time t to the state at time $t+1$. A describes the deterministic part of the underlying process dynamics. The dynamics can be obtained by discretizing the partial differential equation that governs the evolution of the process over time in both time and space, or by some physical rule that governs the behaviour of the process.

B : A matrix of size $(n \times p)$ that contains the stochastic dynamics of the underlying process.

$\underline{w}(t)$: A vector of size $(p \times 1)$ that represents the process noise. This is assumed to be a white sequence.

The linear dynamic measurement model is described by

$$\underline{y}(t) = C(t)\underline{x}(t) + \underline{v}(t) \;\; \underline{v} \sim \mathcal{N}(0, R(t)) \tag{2.22}$$

where,

$\underline{y}(t)$: A vector of size $(m \times 1)$ representing sampled measurements for the process states at time t.

$C(t)$: A matrix of size $(m \times n)$ that indicates which states are measured or the relationship between measurements and the process states.

$\underline{v}(t)$: A vector of size $(m \times 1)$ which is the measurement error.

$R(t)$ is an $m \times m$ matrix, the covariance of the measurement noise $\underline{v}(t)$.

The assumed statistics for the standard Kalman filter are given by

$$E[\underline{w}(t)\underline{w}^T(k)] = \begin{cases} Q(t) & t = k \\ 0 & t \neq k \end{cases} \tag{2.23}$$

$$E[\underline{v}(t)\underline{v}^T(k)] = \begin{cases} R(t) & t = k \\ 0 & t \neq k \end{cases} \tag{2.24}$$

$$E[\underline{w}(t)\underline{v}^T(k)] = 0 \text{ for all } k \text{ and } t \tag{2.25}$$

$$E[\underline{x}(0)\underline{v}^T(k)] = 0 \text{ for all } k \tag{2.26}$$

$$E[\underline{x}(0)\underline{w}^T(k)] = 0 \text{ for all } k \tag{2.27}$$

where (2.25), (2.26) and (2.26) imply the statistical independence[1] of the process noise and the measurement noise, the independence of the state and the measurement noise, and the independence of the state and the process noise, respectively. Several aspects of the filter make it optimal [69]:

- In contrast to static estimation, the effect of the uncertainty in the initial conditions decays by accumulating more measurements over time.
- It processes all the available observations available at different times $t_1, t_2, \cdots, t_T$ regardless of their precision. If an observation has a high noise variance then the filter will associate less weight to it and vice versa.

[1] In the case of uncorrelated Gaussian random variables this implies their independence, however, this is not true for other distributions

- The estimates are computed based on all the available statistics: the process model noise, the measurement noise, and the process prior statistics
- It incorporates the available initial conditions such as the process mean and covariance in the computation of the estimates and error statistics.
- It carries the uncertainty knowledge of the previous estimates and the previous estimates to the next time step, which implies that it does not require all previous data to be stored and then reprocessed every time a new measurement becomes available.

The filter is initialized by the initial conditions of the process, i.e., the process mean m_0 and its covariance P_0 given formally by (2.20) and (2.21).

The Kalman filter consists of two main steps: A prediction step and an update step. The prediction step for both the state estimate and the associated estimation error covariance is given by

$$\hat{\mathrm{x}}(t|t-1) = \mathrm{A}\hat{\mathrm{x}}(t-1|t-1) \tag{2.28}$$

$$\tilde{P}(t|t-1) = A\tilde{P}(t-1|t-1)\mathrm{A}^T + \mathrm{BB}^T \tag{2.29}$$

In the prediction step, the filter computes estimates of the state $x(t)$ based on the model and the previous updated estimates $\hat{x}(t-1|t-1)$. Both the deterministic and stochastic dynamics represented by the model parameters A, B are involved in the prediction step to obtain the best estimates and the associated predicted estimation error covariance $\hat{x}(t|t-1), \tilde{P}(t|t-1)$.

The predicted error covariance $\tilde{P}(t|t-1)$ is computed via (2.29) which is called the discrete time Lyapunov equation.

In the update step, the predicted estimates are combined with the new available mea-

surements at time (t) to obtain estimates based on all the data available up to time (t):

$$\mathrm{K}(t) = \tilde{P}(t|t-1)\mathrm{C}^T(\mathrm{C}\tilde{P}(t|t-1)\mathrm{C}^T + \mathrm{R})^{-1} \quad (2.30)$$

$$\tilde{P}(t|t) = \tilde{P}(t|t-1) - \mathrm{K}(t)\mathrm{C}\tilde{P}(t|t-1) \quad (2.31)$$

$$\hat{\mathrm{x}}(t|t) = \hat{\mathrm{x}}(t|t-1) + \mathrm{K}(t)(\mathrm{y}(t) - \mathrm{C}\hat{\mathrm{x}}(t|t-1)) \quad (2.32)$$

Where $K(t)$ is the gain matrix.

An important feature of the Kalman filter is its ability to compute estimates for the state and the associated estimation error at time t even if no measurements are available at that time. Therefore, in this case, updated estimates are equal to the predicted estimates computed by (2.28) and (2.29). The quality of the predicted estimates depends on the accuracy of the dynamic model parameters i.e., A and B in (2.19). Observe that the error prediction step adds more uncertainty to the updated error covariance $\tilde{P}(t|t)$. The increase in the uncertainty depends on the process includes noise covariance Q. On the other hand, once a measurement becomes available, the filter updates the state estimates, in addition the estimation error is reduced. The amount of reduction depends on how much weight the filter puts on the measurement which depends on the accuracy of the measurement.

Smoothed estimates at anytime t based on all the available measurements can be obtained by applying the Rauch-Tung-Striebel RTS [71] smoothing algorithm. The RTS algorithm consists of two passes: **a forward recursive pass** (left-to-right in Figure 2.2) which is the Kalman filter and **a backward recursive pass** (right-to-left in Figure 2.2) which is the RTS algorithm. The filter is initialized with the smoothed values obtained from the forward pass (i.e., $\hat{x}(T|T)$ and $\tilde{P}(T|T)$). At time t the filter computes smoothed estimates $\hat{x}^s(t|T)$ and $\tilde{P}^s(t|T)$ based on all available measurements up to time T. The smoothed estimates are given by:

$$\Gamma(t) = \tilde{P}(t|t)A^T\tilde{P}^{-1}(t+1|t) \quad (2.33)$$

$$\hat{\underline{x}}^s(t|T) = \hat{\underline{x}}(t|t) + \Gamma(t)\underline{w}(t) \quad (2.34)$$

$$\underline{w}(t) = \hat{\underline{x}}^s(t+1|T) - \hat{\mathrm{x}}(t+1|t) \quad (2.35)$$

$$\tilde{P}(t|T) = \Gamma(t)l(t)\Gamma(t)^T \quad (2.36)$$

$$l(t) = \tilde{P}^s(t+1|T) - \tilde{P}(t+1|t) \quad (2.37)$$

The above form of the RTS smoother is called *fixed-interval* smoothing. In which the measurement time interval is fixed and one can compute smoothed estimates for some or all of the internal points. Two other forms of the smoother exit: *fixed-point smoothing* and *fixed-lag smoothing*. In the former, smoothed estimates are only computed for a single point. In fixed-lag smoothing, smoothed estimates are computed for a fixed length of time back in the past [2, 71].

2.3.2 The information filter

There exists several alternative forms of the Kalman filter that are algebraically equivalent to the standard form presented in section Section 2.3.1. The choice of any specific form depends on the underlying application. One important form of the Kalman filter is the information filter [65, 69]. The information filter propagates the inverse of the estimation error covariance instead of the covariance matrix. This gives the filter a unique feature in that it allows starting up the filter with $P^{-1}(0|-1)$ which is called the information matrix. In many applications, the prior statistical knowledge about the state initial conditions may not be available. This can be modeled by making $P(0|-1) = \infty I$ or equivalently making $P^{-1}(0|-1) = 0I$ which is not allowed in the standard form.

Another advantage of employing the information filter is that it allows exploiting the sparsity of the inverse of the covariance matrix. In case of Markov random fields (MRF), there is a considerable structure in the process covariance. The process covariance of a MRF is generally full but its inverse is sparse and banded [16, 50]. This allows exploiting banded matrix storage and inversion techniques for the filter update step. For large-scale

problems, like remote sensing applications, it makes sense to express the error statistics in terms of the inverse of the error covariance. This will result in a considerable saving in both computational and storage demands.

The information filter differs from the standard filter in the computation of both the updated estimation error and the gain matrix. The updated error is computed by

$$\tilde{P}^{-1}(t|t) = \tilde{P}^{-1}(t|t-1) + C^T R^{-1} C \tag{2.38}$$

Observe that the updated error in this case is computed without computing the gain $K(t)$. The gain in this form is computed *after* the updated error $\tilde{P}(t|t)$ computation via

$$K(t) = \tilde{P}(t|t) C^T R^{-1} \tag{2.39}$$

Note that in order to compute the gain and propagate the estimation error covariance, two $(n \times n)$ matrix inversions (where n is the state dimension) are required at any single time step which is of high computational demand. However, in many applications, a special structure for the filter's matrices exists or can be asserted and this leads to highly efficient storage and reduced computational demands.

2.3.3 Steady-state filter

When all of the Kalman filter's model quantities are time independent, i.e., the model matrices A, B, C, and the noise statistics R, Q and the process statistics are stationary, then the system is called a time-invariant and stationary system. This leads to an important property of the Kalman filter in that the filter itself becomes a time-invariant system and converges to steady-state after some transient duration [71]. At steady state, the filter gain becomes constant i.e., $\lim_{t\to\infty} K(t) = K_{ss}$, where K_{ss} is the gain at steady-state. This leads to constant predicted $\tilde{P}_p$ and updated $\tilde{P}_u$ error matrices. In other words, at steady-state, $\tilde{P}(t+1|t) = \tilde{P}(t|t-1)$, and $\tilde{P}(t+1|t+1) = \tilde{P}(t|t)$. Equation (2.29)

will converge to the predicted estimation error covariance at steady-state $\tilde{P}_p$ and equation (2.31) will converge to the updated estimation error covariance at steady-state $\tilde{P}_u$.

$$\tilde{P}_p = A\tilde{P}_pA^T + Q - AKC\tilde{P}_pA^T \quad (2.40)$$

$$\tilde{P}_u = (I - KC)\left\{A\tilde{P}_uA^T + Q\right\} \quad (2.41)$$

where equation (2.40) is the *standard form* of the Algebraic Riccati equation for the predicted error covariance $\tilde{P}_p$ and (2.41) is the *standard form* of the Algebraic Riccati equation [65] for the updated error covariance $\tilde{P}_u$.

The Kalman filter itself is a solver for $\tilde{P}_p$ or $\tilde{P}_u$. This is because the filter is an iterative procedure to solve both (2.40) and (2.41). One can also solve both (2.40) and (2.41) for $\tilde{P}_p$,$\tilde{P}_u$, respectively by other iterative methods [71].

Given the solution of the predicted error matrix at steady-state $\tilde{P}_p$, the steady-state filter is given by

$$K_{ss} = \tilde{P}_pC^T(C\tilde{P}_pC^T + R)^{-1} \quad (2.42)$$

$$\hat{\underline{x}}(t|t) = \hat{\underline{x}}(t|t-1) + K_{ss}(\mathrm{y}(t) - C\hat{\underline{x}}(t|t-1)) \quad (2.43)$$

$$\hat{\underline{x}}(t+1|t) = \mathrm{A}\hat{\underline{x}}(t|t) \quad (2.44)$$

The steady-state gain K_{ss} is computed only once. Then it is used in (2.43) at each time step to update the predicted state estimate computed by (2.44).

It is important to mention that the transient duration of the time-invariant filter is not affected by the initial prior $P(0|-1)$. $P(0|-1)$ affects only the magnitude of the transient state estimates and the associated error statistics. On the other hand, changing the process noise covariance Q or the measurement noise covariance R does affect the transient duration [69]. If the ratio Q/R is large, then steady-state is reached quickly because the uncertainty in the state and error prediction steps is large compared to the accuracy of the measurements. As a result, the rate of growth of $P(t)$ will increase and

	Computational demand	Storage demand
State prediction	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Error prediction	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
State update	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
Error update	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$

Table 2.1: Summary of the standard Kalman filter computational and storage requirements for each step

therefore, the filter gain will also increase. This implies that the new estimates are heavily dependent upon the new available observations and not upon the prior from the previous time step.

2.3.4 Computational and storage demands of the Kalman filter

Despite the fact that the Kalman filter is a robust recursive estimator, it is limited by the computational and storage demands that depend on the dimensionality of the state vector. As the number of states increases, the filter becomes incapable of dealing with the large computational and storage demands. For 2-D estimation problems, the state vector is of size $n = N \times N$ where N is the length of the field. Table 2.1 summarizes the computational and storage demands for each step in the filter

2.3.5 Approaches for large-scale dynamic estimation

As mentioned in the previous section, the Kalman filter is hindered by computational and storage demands at each time step. Given that most remote sensing applications require a state vector of very high dimensionaliy, most conventional computing facilities are incapable of dealing with the filter's computational and storage requirements.

Several approaches have been investigated to deal with the above limitations of the filter. Following is a brief listing.

A: **Using the information filter and approximate inversion** [16, 45, 69]

In the standard Kalman filter each step requires inverting a matrix of size $m \times m$ where m is the dimensionality of the measurements vector. In some cases, the state vector is much smaller than the measurement vector. In this situation, it is recommended to use the information filter which requires inverting a matrix of size $n \times n$ where n is the number of states to be estimated.

Another approach used in [16], is based on using the information filter and asserting a banded structure for the estimation error covariance matrix. This leads to a huge reduction in both the computational and storage demands of the filter. An $L \times L$ banded matrix that has K bands can be easily stored in a smaller matrix of size $K \times L$. Also in order to avoid the brute-force matrix inversion in the information filter, matrix inversion by polynomial approximation, given in equation (2.16), is used. Substantial savings in both storage and computational demands are gained by employing this approach. This approach has some limitations in the sense that it imposes a nearest-neighbor correlation structure for the underlying estimation error covariance. In addition, the accuracy of matrix inversion by polynomial approximation is highly dependent on the diagonal elements of the matrix to be inverted. If the matrix to be inverted is not diagonally dominant then the polynomial approximation approach is inaccurate.

B: **Sequential processing approach** [45, 46, 63]

In many estimation problems, the measurements obtained at any particular time $y(t)$ are statistically independent. This means that the measurement noise covariance matrix $R(t)$ is diagonal i.e.,

$$E\left\{\underline{v}^{(i)}(t)[\underline{v}(t)^{(k)}]^T\right\} = \begin{cases} R^{(i)} & i = k \\ 0 & i \neq k \end{cases} \tag{2.45}$$

The measurement model (2.22) can be written as

$$\underline{y}(t) = \begin{bmatrix} y^1(t) \\ y^2(t) \\ \vdots \\ y^q(t) \end{bmatrix} = \begin{bmatrix} C^1(t) \\ C^2(t) \\ \vdots \\ C^q(t) \end{bmatrix} \underline{x}(t) + \begin{bmatrix} v^1(t) \\ v^2(t) \\ \vdots \\ v^q(t) \end{bmatrix} \tag{2.46}$$

Observe that in the above formulation we have q statistically independent measurements each of them can be a scalar or of size $m^{(i)} \times 1$. This makes

$$\sum_{i=1}^{q} m^{(i)} = m$$

since the measurements at time t are independent, they can be processed individually and independently. Consequently, the huge matrix inversion in the gain matrix computation in equation (2.30) can be avoided. Instead of inverting an $m \times m$ matrix simultaneously, one must compute q smaller matrix inversions at any time step. Furthermore, if all measurements are independent, i.e., $m^{(i)} = 1$ for all $i = 1, 2, \cdots, q$, then no matrix inversions are required at all. The update step for the Kalman filter is computed iteratively as follows

$$K^{(i)}(t) = (C^{(i)}(t)P^{(i-1)}(t|t)C^{(i)T}(t)))^{-1}P^{(i-1)}(t|t-1)C^{(i)T}(t) \tag{2.47}$$

$$P^{(i)}(t|t) = P^{(i)}(t|t) - K^{(i)}(t) = P^{(i)}(t|t)C^{(i)T}(t) \tag{2.48}$$

$$\underline{\hat{x}}^{(i)}(t) = \underline{\hat{x}}^{(i-1)}(t) + K^{(i)}(t)\left[y^{(i)}(t) - C^{(i)}(t)\underline{\hat{x}}^{(i-1)}(t)\right] \tag{2.49}$$

The above steps are repeated for $i = 1, 2, \cdots, q$.

The above approach highly reduces both the computational demands for matrix inversion and also the numerical errors involved in inverting large matrices. However, it does not handle the huge storage demand required for storing the error estimation covariance matrices. Also it does not deal with the computational demand involved in

matrix multiplication which is necessary in the prediction step. It is worth mentioning that there exists an equivalent sequential processing formulation for the information filter [46, 63]

B: **Reduced order approach** [65, 71]

This approach aims towards reducing the dimensionality of the state vector by decoupling the original dynamic system into smaller subsystems that are processed independently. By doing so, we actually divide the large estimation problem into many small estimation problems that can be solved easily with computational and storage demands depending on the size of the state vector for each subsystem. Consider the dynamic models given in equation (2.19),(2.22), if the models parameters A, C, Q, R are block diagonal, then the dynamic models can be written as

$$\begin{bmatrix} \underline{x}_1(t+1) \\ \underline{x}_2(t+1) \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} + \begin{bmatrix} \underline{w}_1 \\ \underline{w}_2 \end{bmatrix} \tag{2.50}$$

$$\begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} + \begin{bmatrix} \underline{v}_1 \\ \underline{v}_2 \end{bmatrix} \tag{2.51}$$

For the above example, the computational and storage demands for the filter per time step are reduced to $\mathcal{O}(n_1^3 + n_2^3)$ instead of $\mathcal{O}(n_1 + n_2)^3$.

On the other hand, if the filter matrices are not block diagonal then asserting a block diagonal structure may lead to substantial approximation depending on the underlying problem.

A reduced order filter can also be implemented in the case of perfect measurements [71]. A set of perfect measurements (i.e., $R = 0I$) reduces the number of related states to be estimated. If k perfect measurements exist, then instead of estimating a state vector of length n, one needs to deal with only a state vector of size $n - k$. This approach reduces the storage and computational demands of the filter. However, in most cases,

the k perfect measurements are usually linearly related to the n states. This issue requires adjusting the estimator steps. The reader is referred to [71] for a more detailed discussion on this issue.

Standard applications of the reduced order filter appears in works related to 2-D image restoration problems. Extensive research exists on Kalman filtering for 2-D image restoration [3, 5, 9, 13, 59, 100]. Generally, the filter in this case is called a reduced update Kalman filter. In order to reduce the computational demands when dealing with the 2-D image, the restoration problem is based on a dynamic model for the image that considers only a small neighbourhood as the state vector. The small local dynamic model switches throughout the image. the dynamic model is given by [59]

$$x(m,n) = \sum_{(k,l)\in\omega} a(k,l)x(m-k,n-l) + w(m,n) \tag{2.52}$$

where $a(k,l)$ are the model parameters, and $w(m,n)$ is a zero-mean white Gaussian noise. The set ω defines those pixel locations which are used to define the autoregressive model of the image. The set ω is given by

$$\omega = \{(m-1,n),(m-1,n-1),(m,n-1),(m+1,n-1)\} \tag{2.53}$$

The appropriate model parameters for each region need to be estimated. The 2-D Kalman filtering deals with solving a static estimation problem (i.e., image restoration) using a dynamic estimation method. This method is not applicable to large-scale dynamic estimation where at each time new 2-D frame (image) data arrives and needs to be considered to update the previous estimates.

2.4 Testing the dynamic estimator performance

The standard Kalman filter propagates the error statistics through time by (2.29). Then, the gain matrix $K(t)$ is computed based on the propagated error covariance $\tilde{P}(t+1|t)$. In large scale dynamic estimation problems, due to the huge computational demands, exact error propagation is impossible. Alternatively, the error is propagated approximately. In this case, the gain $K(t)$ computed based on the approximate error covariances is no longer optimal and it is called the suboptimal gain and hence the filter is called a suboptimal filter [2, 65].

In the case of a suboptimal filter, the error statistics produced by (2.29), (2.31) are *not the actual* error statistics, since the suboptimal filter does not use the optimal gain $K(t)$ in its computation.

To be able to compute the actual error statistics, we need to look at the temporal dynamics of the updated and predicted estimation errors of the Kalman filter.

The general equations that govern the propagation of the estimation error statistics for any linear recursive estimator are derived from the estimation error temporal dynamics [50]. The updated and predicted dynamics of the estimation error are given by

$$e(t|t) = (I - K(t)C)e(t|t-1) - K(t)v(t) \quad (2.54)$$

$$e(t+1|t) = Ae(t|t) + w(t) \quad (2.55)$$

where $w(t) \sim \mathcal{N}(0, Q)$ and $v(t) \sim \mathcal{N}(0, R)$ are the process noise and the measurement noise, respectively. $e(t|t)$ is the error in computing the updated state estimates $\hat{\underline{x}}(t|t)$ and $e(t|t-1)$ is the error in the state prediction step resulting from computing the states estimates using (2.28).

Hence, from (2.55),(2.54), the updated error covariance $\tilde{P}(t|t)$ and the predicted error covariance $\tilde{P}(t+1|t)$ of the suboptimal filter are propagated by

$$\tilde{P}(t|t) = (I - K(t)C)\tilde{P}(t|t-1)(I - K(t)C)^T + K(t)RK(t)^T \quad (2.56)$$

$$\tilde{P}(t+1|t) = A\tilde{P}(t|t)A^T + Q \tag{2.57}$$

Now, the above two equations give the same answer as (2.29),(2.31) only in the case of the optimal Kalman filter with gain $K(t)$ computed based on actual propagated error statistics. However, in the case of propagating or computing the error statistics approximately, then the above equations will give the actual error statistics, but (2.29),(2.31) will give the error statistics that the filter believes and uses in its recursion.

Combining both equations (2.29) and (2.31) together gives a single dynamic equation that governs the temporal propagation of the predicted estimation error covariance from $\tilde{P}(t|t-1)$ to $\tilde{P}(t+1|t)$.

$$\tilde{P}(t+1|t) = A\tilde{P}(t|t-1)A^T + Q - AK(t)C\tilde{P}(t|t-1)A^T \tag{2.58}$$

Observe that equation (2.58) gives the dynamics of the predicted estimation error statistics $\tilde{P}(t|t-1)$. An alternative form for the dynamics that propagate the updated estimation error statistics $\tilde{P}(t|t)$ to $\tilde{P}(t+1|t+1)$ is given by

$$\tilde{P}(t+1|t+1) = (I - K(t+1)C)\left\{A\tilde{P}(t|t)A^T + Q\right\} \tag{2.59}$$

Similarly, combining equations (2.56), and (2.57) together gives an alternative form to the standard Riccati equation (2.58) which is called the *Joseph stabilized form* [65] of the Riccati equation that governs the temporal dynamic of the predicted estimation error $\tilde{P}(t+1|t)$

$$\begin{aligned}\tilde{P}(t+1|t) &= A\Big\{(I - K(t)C)\tilde{P}(t|t-1)(I - K(t)C)^T \\ &+ K(t)RK(t)^T\Big\}A^T + Q\end{aligned} \tag{2.60}$$

Similarly, one can have the *Joseph stabilized form* to (2.60) for the dynamics that propa-

gate the updated estimation error covariance through time from $\tilde{P}(t|t)$ to $\tilde{P}(t+1|t+1)$

$$\begin{aligned}\tilde{P}(t+1|t+1) &= (I-K(t+1)C)A\tilde{P}(t|t)A^T(I-K(t+1)C)^T \\ &+ (I-K(t+1)C)Q(I-K(t+1)C)^T \\ &+ K(t+1)RK(t+1)^T \end{aligned} \tag{2.61}$$

For a stationary and time-invariant dynamic systems, (2.60) will converge to the predicted estimation error covariance at steady-state $\tilde{P}_p$ and (2.61) will converge to the updated estimation error covariance at steady-state $\tilde{P}_u$. The steady-state expressions for both (2.60) and (2.61) are given by

$$\tilde{P}_p = A\left\{(I-KC)\tilde{P}_p(I-KC)^T + KRK^T\right\}A^T + Q \tag{2.62}$$

$$\begin{aligned}\tilde{P}_u &= (I-KC)A\tilde{P}_uA^T(I-KC)^T \\ &+ (I-KC)Q(I-KC)^T + KRK^T \end{aligned} \tag{2.63}$$

where equation (2.62) is the *stabilized form* of the Algebraic Riccati equation for the predicted error covariance $\tilde{P}_p$ and (2.63) is the *stabilized form* of the Algebraic Riccati equation for the updated error covariance $\tilde{P}_u$.

Based on the above, in order to be able to assess a suboptimal estimator, several criteria exist and the selection between them is application dependent:

- If one is looking for a good estimator then, the estimates computed by the the approximate filter and the estimates computed by the standard filter should be compared.
- If the resulting error statistics are important then, the error statistics produced by the approximate filter based on (2.29), (2.31) and the actual error statistics computed by (2.56),(2.57) should be compared.
- Alternatively, the approximate error statistics produced by the approximate filter

(based on the approximate gain) are compared to the true error statistics produced by the standard Kalman filter based on the true gain.

- If the behaviour of the approximate filter at steady-state is to be studied, then the solutions of equations (2.62),(2.63) based on the approximate gain should be compared to the solutions of (2.40), (2.41).

In general, testing the performance of a suboptimal filter depends on many related factors. A suboptimal filter that performs well with respect to a specific application may not have the same performance when applied to another application. Consequently, in almost all works related to the development of an approximate filter, the performance is compared to the *exact solution* obtained by the standard Kalman filter. For large-scale problems comparison is based on small size problems [61].

Chapter 3

Multiscale Dynamic Estimation

This chapter is devoted to two subjects: multiscale estimation theory and multiscale-based dynamic estimation.

3.1 Challenges in large-scale dynamic estimation

Large-scale dynamic estimation problems appear in many scientific applications: image processing [9, 16] and remote sensing [11, 12, 23], to name a few. As mentioned in Table 2.1, as the number of states gets larger, the computational and storage demands of the conventional dynamic estimation tool, i.e., the Kalman filter become infeasible. In Section 2.3.5, several approaches to large-scale dynamic estimation problems have been presented. The performance of these approaches is problem dependent. These approaches do not directly deal with the issue of the problem. The sequential processing approach is applied only when measurements at any time step are statistically independent. This approach could only reduce the computational demand of the matrix inversion in the update step. The prediction step is done exactly which means that the full updated error covariance has to be available. Another approach is based on employing the information filter with the local correlation assumption (banded error covariance matrices).

This approach reduces the storage demand of the filter. However, matrix inversion, in this approach, is based on polynomial approximation which works only for diagonally dominant matrices. A reduced order filter is based on partitioning the system dynamics into smaller subsystems. Substantial approximation arises if the original system is not separable.

More recently, an efficient multiscale static estimation algorithm that can deal with large-scale static estimation problems has been developed [17, 67] and employed for dynamic estimation problems [50]. The following sections present details of the multiscale framework.

3.2 Overview on Multiscale estimation

As discussed in Section 2.2.4, there are several techniques that can be used to attack the high computational effort involved in LLSE. However, these techniques have their limitations. As an example, the multigrid method, which is one of the efficient iterative methods that can provide estimates with a fast convergence rate . However, computing the error statistics can only be done by brute-force and is considered computationally inefficient for large scale problems. The FFT technique is another method for solving large-scale static estimation problems but requires that the random process and the measurements noise be stationary, in addition, to the requirement of point measurements. These conditions are not usually satisfied in large-scale remote sensing problems like the one that will be investigated in this research.

It is well known that when dealing with large scale problems, one can try to find an approximate solution by subdividing the problem into smaller ones which are easy to solve or by creating another similar approximate problem which can be solved exactly [88]. The second option is the one adapted in the multiscale framework [32]. The idea is motivated by the success of multiresolution analysis which combats the computational demands of a large-scale problem by solving coarse versions that are computa-

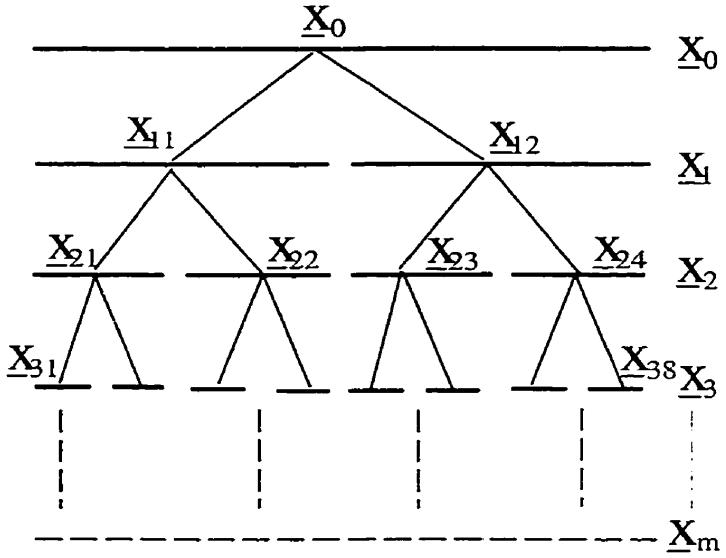


Figure 3.1: An example to illustrate the hierarchical decomposition of a 1-D random process on a dyadic tree.

tionally simpler and then uses these coarse versions to guide and speed up their corresponding higher resolution parts. Recall from the previous chapter that in the standard Bayesian LLSE, the prior knowledge of the statistical structure of a random process $\underline{x}$ is explicitly given in a form of the covariance matrix $P_x = E[\underline{x}\underline{x}^T]$. Bayesian LLSE depends on the prior covariance to compute estimate equations (2.12),(2.14) which is infeasible in the case of large-scale estimation problems. Multiscale estimation [32], on the other hand, is based on modeling the statistical structure of the underlying stochastic process at multiple scales on a tree as depicted in Figure 3.1. This will lead to a scale-to-scale relationship and consequently allow an efficient estimation solution for large-scale problems. The goal is to get an approximate version of the whole random field statistics P_x (the desired prior covariance) at the finest scale in a finite number of scales referred to by M. In doing so, the random field is modeled on a tree with coarser scales toward the top of the tree. The finest representation of the random field will approximately represent

the statistical features of the original random field:

$$P_{\text{finest scale}} = P_{x^M} \approx P_x$$

To illustrate the above [33], consider a large 1-D process or a 2-D static random field, stacked as one vector $\underline{x}$, with a prior covariance P_x and the linear measurements model given by $\underline{y} = C\underline{x} + \underline{v}$. The random 1-D process is modeled on a dyadic tree as shown in Figure 3.1. The states $\underline{x}$ at each level capture part of the statistical structure of the original random field $\underline{x}$. More detailed discussion of the multiscale modeling will be given in Section 3.4. At coarsest level $m = 0$, the field is approximated by a single state vector:

$$\underline{x}_0$$

At the next level $m = 1$, the random field is approximated by:

$$\underline{x}_1 = \begin{bmatrix} \underline{x}_{11} \\ \underline{x}_{12} \end{bmatrix}$$

At the third level $m = 2$, the process is approximated by:

$$\underline{x}_2 = \begin{bmatrix} \underline{x}_{21} \\ \underline{x}_{22} \\ \underline{x}_{23} \\ \underline{x}_{24} \end{bmatrix} \tag{3.1}$$

It is obvious that as we increase the scale (moving down in the tree), we add more details to the representation of the statistical structure of the random field. At the finest scale $m = M$ we capture approximately the whole statistical structure of the random process.

The basic model that governs the relationship between states at different scales is

given by the first-order Markov model:

$$\underline{x}_i = A_i \underline{x}_{i-1} + B_i \underline{w}_i \tag{3.2}$$

where $\underline{w}$ is Gaussian zero-mean with identity covariance, white noise process uncorrelated with $\underline{x}_0$

By using the recursive model given in (3.2), one can express the relation between states at level 2 and level 1 by:

$$\begin{aligned} \underline{x}_2 &= \begin{bmatrix} \underline{x}_{21} \\ \underline{x}_{22} \\ \underline{x}_{23} \\ \underline{x}_{24} \end{bmatrix} = A_2 \underline{x}_1 + B_2 \underline{w}_2 \\ &= \begin{bmatrix} A_{21} & & \\ A_{22} & & \\ & A_{23} & \\ & A_{24} & \end{bmatrix} \begin{bmatrix} \underline{x}_{11} \\ \underline{x}_{12} \end{bmatrix} + \begin{bmatrix} B_{21} & & & \\ & B_{22} & & \\ & & B_{23} & \\ & & & B_{24} \end{bmatrix} \begin{bmatrix} \underline{w}_{21} \\ \underline{w}_{22} \\ \underline{w}_{23} \\ \underline{w}_{24} \end{bmatrix} \end{aligned} \tag{3.3}$$

Equation (3.3) demonstrates that by modeling the random field on the tree, we break down the dynamics of the states at each level as they evolve from scale-to-scale. As an example, the state $\underline{x}_1$ at level 1 is composed of four smaller states each with its own model parameters A_{ij}, B_{ij}. This breakdown of the relationships between the tree nodes is vital in the sense that it allows breaking down the large estimation problem into smaller estimation problems that can be solved efficiently. The following sections will give a formal description of multiscale processing and modeling [17, 32, 50, 53, 67].

3.3 Multiscale Processing

In the multiscale framework, the random process is modeled on a tree structure. In

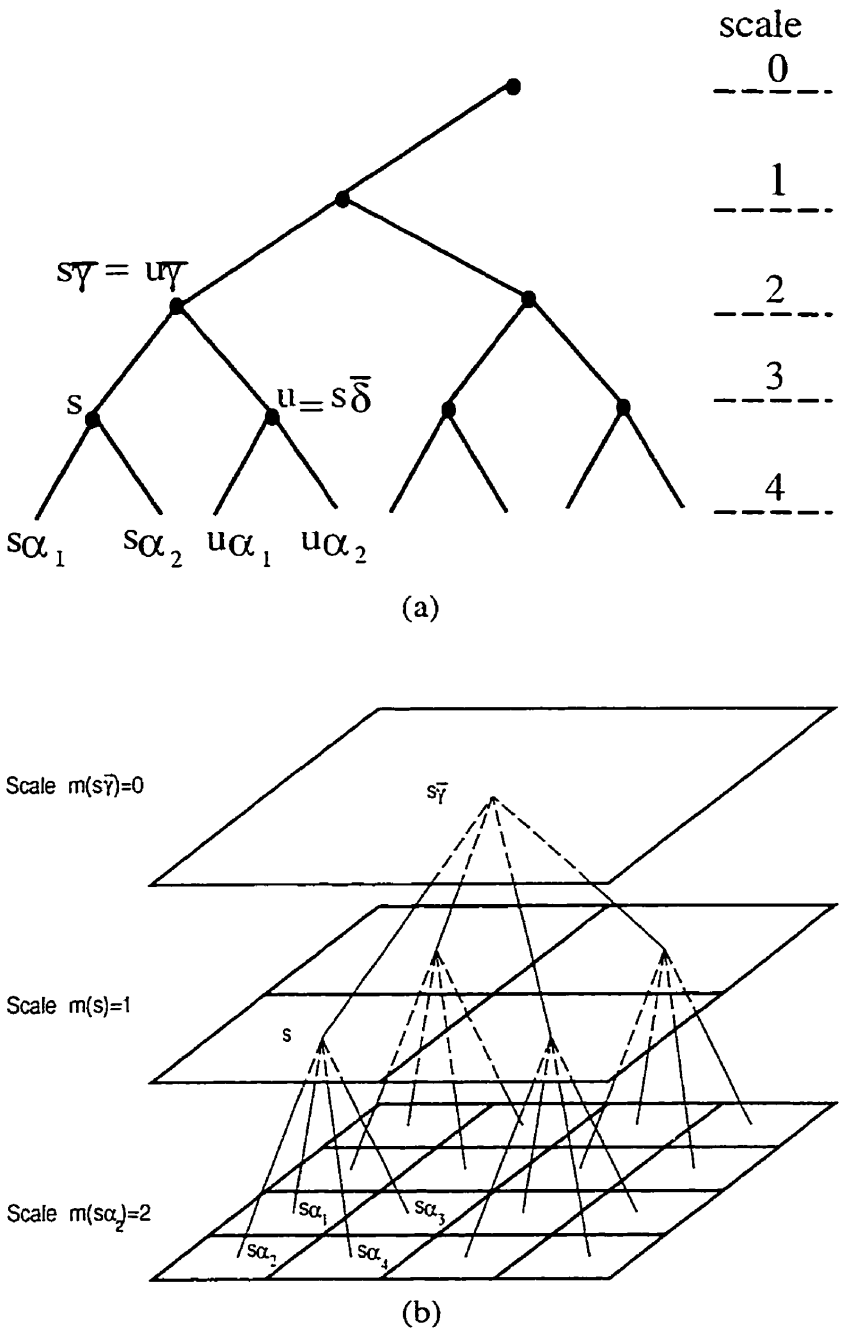


Figure 3.2: Illustration of different types of trees used in the multiscale framework to model 1-D and 2-D stochastic processes [32]. (a) is the multiscale tree used to model 1-D processes and (b) is a quadtree used to model 2-D random fields.

order to generalize the recursive model given by (3.2), any node on the tree is referred to by a variable s, so $x(s)$ refers to state x at node s on the tree and $\bar{\gamma}$ is a raising operator for state s to its parent state $x(s\gamma)$. The general scale-to-scale recursive model for representing the dynamics of the random process on the tree is given by

$$x(s) = A(s)x(s\bar{\gamma}) + B(s)w(s) \tag{3.4}$$

where $w(s)$ is a zero-mean vector with identity covariance, white Gaussian noise process uncorrelated with $x(0)$. $A(s), B(s)$ are deterministic quantities defining the process statistics on the tree. The linear measurement model in the multiscale framework is given by

$$y(s) = C(s)x(s) + v(s) \tag{3.5}$$

where $C(s)$ is a selection matrix defining measured states at node (s), and $v(s)$ is white Gaussian noise. The measurement model is general in the sense that it allows measurements to be available at any scale.

Figure 3.2 (a) is an example of a general dyadic tree which is used to model a 1-D process and Figure 3.2 (b) is a quadtree used to model a 2-D random field. In both cases, the finest scale represents the underlying random process. Referring to previous figures, the multiscale tree is characterized, mainly, by two parameters: $q(s)$ which represents the descendants of each node s (except those at the finest scale), and M which represents the number of scales in the tree. The scale of any node is given by $m(s)$, where $0 \leq m(s) \leq M$. The finer scales have larger values of $m(s)$. The root node of the tree is denoted by 0 and its scale is $m(s) = 0$. In general, a uniform dyadic tree Figure 3.2 (a) (i.e., each node has two child nodes except at the finest scale) is used to model 1-D random processes and a quadtree (each node has four child nodes except at the finest scale) is used to model a 2-D random field. Referring to Figure 3.2 every node (s) on the tree is connected to a unique parent node, $s\bar{\gamma}$, at the previous coarser level. In ad-

dition, each node s (except those at the finest level) is connected to several child nodes $x(s\alpha_i), (i = 1, \cdot \cdots, q)$ at the next finer level.

The efficiency of the multiscale estimation algorithm is based upon the Markovianity property of (3.4), i.e., conditioned on any node on the tree, each of the $q + 1$ subtrees connected to this node are conditionally decorrelated. The processing of that data in the subtrees below a given node s is independent of each other. This is a key point behind the multiscale framework.

At the root node, the initial conditions are given by

$$E[x(0)] = x(0) \qquad E[x(0)x^T(0)] = P(0) \tag{3.6}$$

The initial conditions $x(0)$ and $P(0)$ represent the coarsest prior representation of the random process at the finest scale x^M. Given the initial conditions and the model parameters $A(s), B(s)$, any entry in the original process covariance matrix can be easily computed. For example, the cross-covariance matrix between the two nodes on the tree $x(s\alpha_1)$ and $x(s\alpha_2)$ in Figure 3.2 (a) is

$$P(s\alpha_1, s\alpha_2) = E[x(s\alpha_1)x^T(s\alpha_2)] = \begin{bmatrix} P(s\alpha_1) & P(s\alpha_1, s\alpha_2) \\ P(s\alpha_1, s\alpha_2)^T & P(s\alpha_2) \end{bmatrix} \tag{3.7}$$

Using the recursive model (3.4) to express both nodes $x(s\alpha_1)$ and $x(s\alpha_2)$ in terms of their model parameters $A(s_1), A(s_2), B(s_1), B(s_2)$ and their common parent state $x(s)$

$$\begin{aligned} x(s\alpha_1) &= A(s\alpha_1)x(s) + B(s\alpha_1)w(s\alpha_1) \\ x(s\alpha_2) &= A(s\alpha_2)x(s) + B(s\alpha_2)w(s\alpha_2) \end{aligned}$$

Then apply the expectation, we get

$$E[x(s\alpha_1)x(s\alpha_2)^T] = A(s\alpha_1)E[x(s)x(s)^T]A(s\alpha_2)^T + B(s\alpha_1)E[w(s\alpha_1)w(s\alpha_2)^T]B(s\alpha_2)^T \tag{3.8}$$

By considering the fact that the noise components are uncorrelated in scale, we get

$$P(s\alpha_1, s\alpha_2) = A(s\alpha_1)P(s)A(s\alpha_2)^T \tag{3.9}$$

The covariance $P(s)$ of state $x(s)$ at any tree node s is computed by the scale-recursive Lyapunov equation

$$P(s) = A(s)P(s\bar{\gamma})A^T(s) + B(s)B^T(s) \tag{3.10}$$

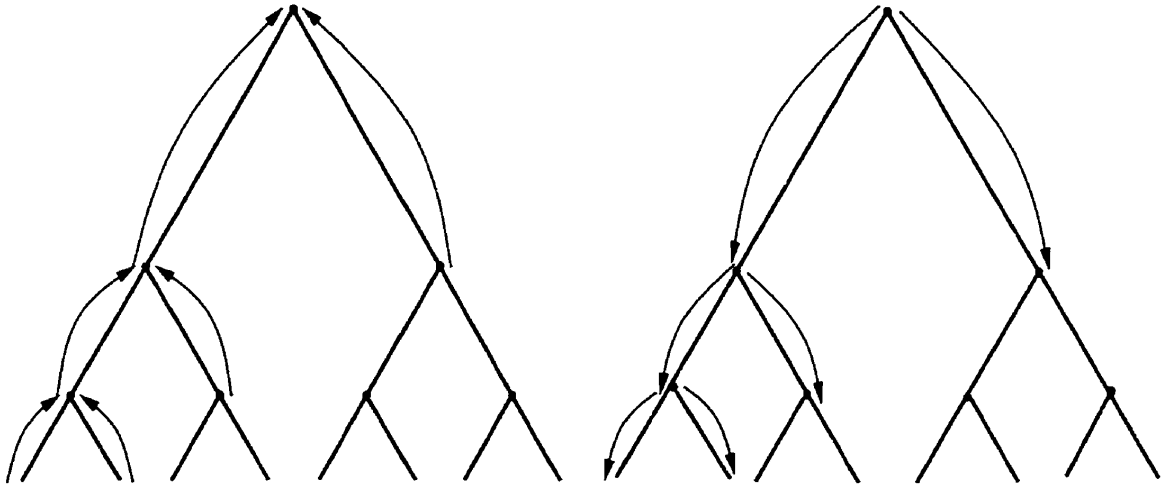
The Markovianity property implied by (3.4), allows efficient estimation based on the RTS smoothing algorithm [71]. Recall from Section 2.3 that the RTS smoothing algorithm involves two steps: a forward Kalman filter sweep and a backward sweep to compute the smoothed estimates. In the forward sweep, best estimates of the states of the process at time t are computed based on all measurements available up to t.

$$\hat{x}(t) = E[x(t)|y(k), k \leq t] \tag{3.11}$$

In the backward sweep, smoothed estimates at time t are obtained based on all available measurements.

$$\hat{x}^s(t) = E[x(t)|y(k), 0 \leq k \leq T] \tag{3.12}$$

The multiscale framework generalizes the RTS smoother by modeling the process on a tree and marching in scale s, instead of time, from fine-to-coarse as an upward (forward) sweep (Figure 3.3 (a)) and from coarse-to-fine as a downward (backward)(figure Figure 3.3 (b)). In the upward sweep, the Kalman filter prediction step computes an estimate of state $x(s)$ at node s and scale $m(s) = j$ based on the measurements available at



a) Upward Pass b) Downward Pass

Figure 3.3: Illustration of the two main steps used in multiscale estimation. The upward pass is the Kalman filter and the downward pass is the RTS smoother.

each of the subtrees $s\alpha_i$ descending from node s individually.

$$\hat{x}(s)_{m=j} = E[x(s)_{m=j}|y(k), k = M, M-1, \cdots, k = j-1] \tag{3.13}$$

Next, the individual predicted estimates of state $x(s)$ are merged by a merge step to obtain the best linear estimates of $x(s)$ based on all the data available at the child nodes $s\alpha_i$. Then, in the update step, the best estimates of $x(s)$ and scale $m(s) = j$ is obtained by involving measurements available at node s.

$$\hat{x}(s)_{m=j} = E[x(s)_{m=j}|y(k), k = M, M-1, \cdots, k = j] \tag{3.14}$$

The downward sweep follows by marching down from the root node $x(0)$ to the finest scale. In this step, the best estimate of $x(s)$ at any node is computed based on available

measurements everywhere on the tree.

$$\hat{x}^s(s)_{m=j} = E[x(s)_{m=i}|y(k), k = 0, M-1, \cdots, k = M] \tag{3.15}$$

The multiscale estimation algorithm differs from the standard Kalman filter and RTS smoother in that the algorithm proceeds from fine-to-coarse (i.e., in the inverse direction of the model given in (3.4),(3.5)). Consequently, a fine-to-coarse model that represents the states $x(s\bar{\gamma})$ in terms of $x(s)$ and a noise term that is uncorrelated with state $x(s)$ is constructed and used in the upward sweep. Another key difference between the multiscale smoother and standard RTS smoother is the presence of a merge step in the prediction step of the standard Kalman filter. As each node s on the tree (except those at the finest scale) has q children, q predictions are obtained. The merge step is necessary to combine all these predicted estimates. The detailed multiscale estimation algorithm is summarized in Appendix A.

3.4 Multiscale Modeling

Recall that we refer to any two random variables x, y as being conditionally uncorrelated if conditioning on another random variable z the cross conditional correlation of x, y becomes the product of the individual conditional mean [92]

$$E[xy|z] = E[x|z]E[y|z] \tag{3.16}$$

The efficiency behind the multiscale framework is based on the conditional decorrelation property (3.16) which allows breaking the large estimation problem into smaller problems that are easier to be solved. The task of the multiscale modeling is to determine the state variables at any tree node that will exactly or approximately satisfy the tree conditional decorrelation assumption. The random process $x(s)$ in the recursive model is assumed to be hierarchical. In other words, the statistical structure of the random pro-

cess can be modeled on multiple resolutions. The model parameters (i.e., $A(s), B(s)$) allow the model given by (3.4) to recursively approximate the state at any tree node $x(s)$ in terms of its parent state $x(s\bar{\gamma})$ and some estimation error term $w(s)$. That is

$$x(s) = E[x(s)|x(s\bar{\gamma})] + w(s) \tag{3.17}$$

To clarify (3.17), observe that if we assume that the parent state $x(s\bar{\gamma})$ represents a measurement for the child state $x(s)$ with covariance $P(s\bar{\gamma})$, then the cross-covariance between the unknown (the child state) and the parent state is $P(s, s\bar{\gamma})$, and applying the LLSE (2.12) leads us to have the following expressions for the multiscale model parameters $A(s), B(s)$ [32]

$$\begin{aligned} A(s) &= P(s, s\bar{\gamma})P^{-1}(s\bar{\gamma}) && (3.18)\\ E[w(s)w^T(s)] &= B(s)B^T(s) && (3.19)\\ &= P(s) - P(s, s\bar{\gamma})P^{-1}(s\bar{\gamma})P^T(s, s\bar{\gamma})P^{-1}(s\bar{\gamma}) && (3.20) \end{aligned}$$

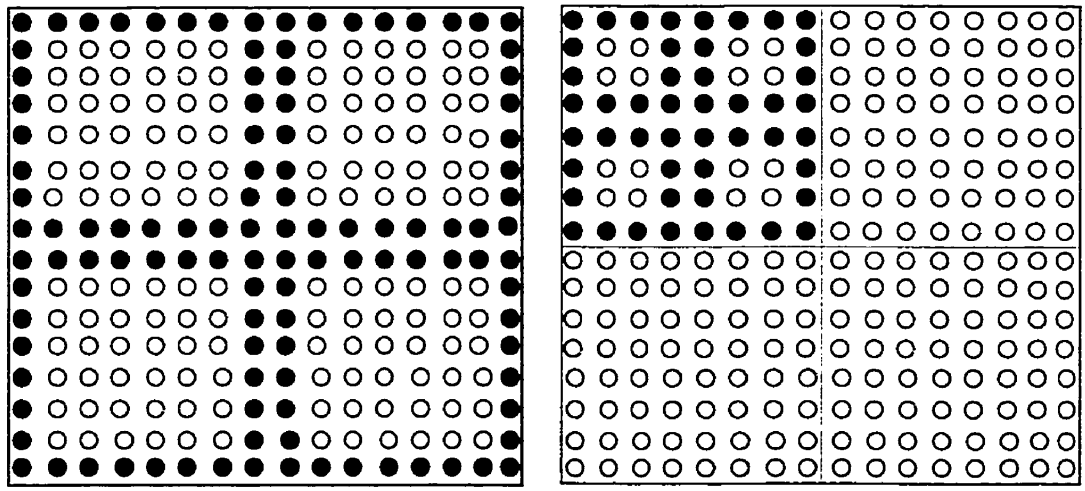
Now the actual multiscale state definition $x(s)$ depends on the nature of the random process. A rich literature is available on the multiscale modeling issue [32, 40, 50, 53, 66].

In general, there are three classes of random processes that have been successfully modeled by the multiscale framework: Markov random fields MRFs [32, 66], general Gaussian random field GRF i.e., processes where all variables are Gaussian, and processes that have power spectra density like $1/f^u$ [32, 103].

As this research is based on multiscale models for 2-D MRFs, a detailed discussion of this type will is covered in the following.

3.4.1 Multiscale models for 2-D MRF

To give a formal definition for discrete 2-D MRFs [21] we should define a neighbour-



(a) (b)

Figure 3.4: One possible arrangement of boundary elements (•) that are kept as multi-scale states for 16 × 16 MRF. Left (a) shows the boundary pixels for the root node $x(0)$ and right (b) Shows the boundary pixels for the left-north quadrant for the next level in the quadtree $x(s\alpha_1)$.

hood system over a rectangular lattice L representing the whole 2-D field. Let $\{N_{ij} \subset L\}$ be a set of field elements $x(k,l) \in L$ located in the vicinity of element $x(i,j)$. Therefore, any element $x(k,l)$ is in the neighbourhood of element $x(i,j)$ if $x(k,l) \in N_{i,j}$. A 2-D process is called MRF if

$$P(x(i,j)|x(k,l),(k,l) \in L) = P(x(i,j)|x(k,l),(k,l) \in N_{ij}) \tag{3.21}$$

MRFs are classified according to the definition of the neighbourhood system. A first-order MRF is associated with a first-order neighbourhood system which consists of the four nearest neighbours of each field element. A second-order MRF is associated with a second-order neighbourhood system which consists of the eight nearest neighbours of each field element. The set of points that are neighbours of a field subset $\{\Omega \subset L\}$ are called the *boundary* of region Ω. The definition of the boundary also varies according to

the neighbourhood order.

Now depending on the order of the MRF, a set of boundary points of a closed region conditionally decorrelates the set of inside points from the rest of the domain. In order to model a 2-D MRF on the multiscale tree we have to define the trees internal states $x(s)$ so that the conditional decorrelation property is satisfied as in the model (3.4). In addition, the realized process covariance at the finest scale of the tree P_{x^M} matches approximately the desired process covariance at the finest scale P_x. From the MRF definition given above [21], the decorrelation property is completely satisfied by making the state vector $x(s)$ at any tree node composed of the boundary elements of its children. That is,

$$x(s) = W(s)x^M \tag{3.22}$$

Equation (3.22) states that the multiscale states $x(s)$ are expressed as linear combinations of the states x^M at the finest scale nodes. where $W(s)$ is a matrix, defined for each tree node, sampling every pixel of the random process x^M along the boundaries of the children of state $x(s)$ as depicted in Figure 3.4. By making $x(s)$ equal to the boundary pixels of its children (i.e., the states $x(s\alpha_i)$ for $(i = 1, \cdots, 4)$), the Markovianity conditional decorrelation property is satisfied. In other words, $x(s)$ is decorrelated from all subtrees below it and the remainder of the tree nodes at the finest scale.

The multiscale model parameters $A(s), B(s)$ require the covariance of the state $x(s)$ and the cross-covariance of $x(s)$ and $x(s\bar{\gamma})$. Given the state definition as in (3.22), the multiscale model parameters $A(s), B(s)$ are then computed according to (3.18),(3.20) as follows:

$$A(s) = [W(s)P_x W^T(s\bar{\gamma})][W(s\bar{\gamma})P_x W^T(s\bar{\gamma})]^{-1} \tag{3.23}$$

$$B(s)B^T(s) = P(s) - A(s)P(s\bar{\gamma})A^T(s) \tag{3.24}$$

Issues concerning the equivalence of (3.18), (3.20) and (3.23), (3.24) can be found in

[40, 50].

A critical issue for multiscale modeling of a 2-D MRF is the growing size of the tree states that is proportional to the field dimensions. For a field of size n, the maximum state dimension (the root state $x(0)$) is proportional to $\sqrt{n}$. Usually, the highest states dimensions occur in the first two levels of the tree. For large-scale estimation problems, like the ocean surface temperature [34], exact multiscale realization of the field statistics requires high computational and storage demands as the tree states required to realize the process get larger. Figure 3.4 illustrates the exact multiscale modeling for a 16×16 2-D MRF. The model is called a *full order multiscale model* in that all the necessary boundary elements are kept as tree states. The computational demand required to do static estimation for processes that are modeled by exact MRF models is $\mathcal{O}(nd^3)$ where $n = N \times N$ and d is the dimension of the largest state vector $x(s)$ on the tree. In case of fixing the root state dimension to N, then the computational complexity becomes $\mathcal{O}(n^{5/2})$

3.4.2 Other Multiscale models

Observe from Figure 3.4 that some of the field elements assigned to the root node state are also assigned to the four descendant states in the next level. This results in some redundant information kept at multiple scales on the tree. Because of such redundancy these models are called redundant models. Another type of multiscale modeling for MRFs is called *non-redundant multiscale model*. In non-redundant models elements of the 2-D field appear only once on the tree [50]. This implies that the 2-D process is no longer mapped to the finest scale but is distributed among all nodes on the tree. An example is depicted in Figure 3.5 (a,b) which display the elements that are kept as states for the root node and the next level for a 17×17 field. The multiscale computational complexity for non-redundant boundary models for 2-D MRF is $\mathcal{O}(n^{3/2})$.

The two previous multiscale realizations of 2-D MRF are exact. In order to avoid

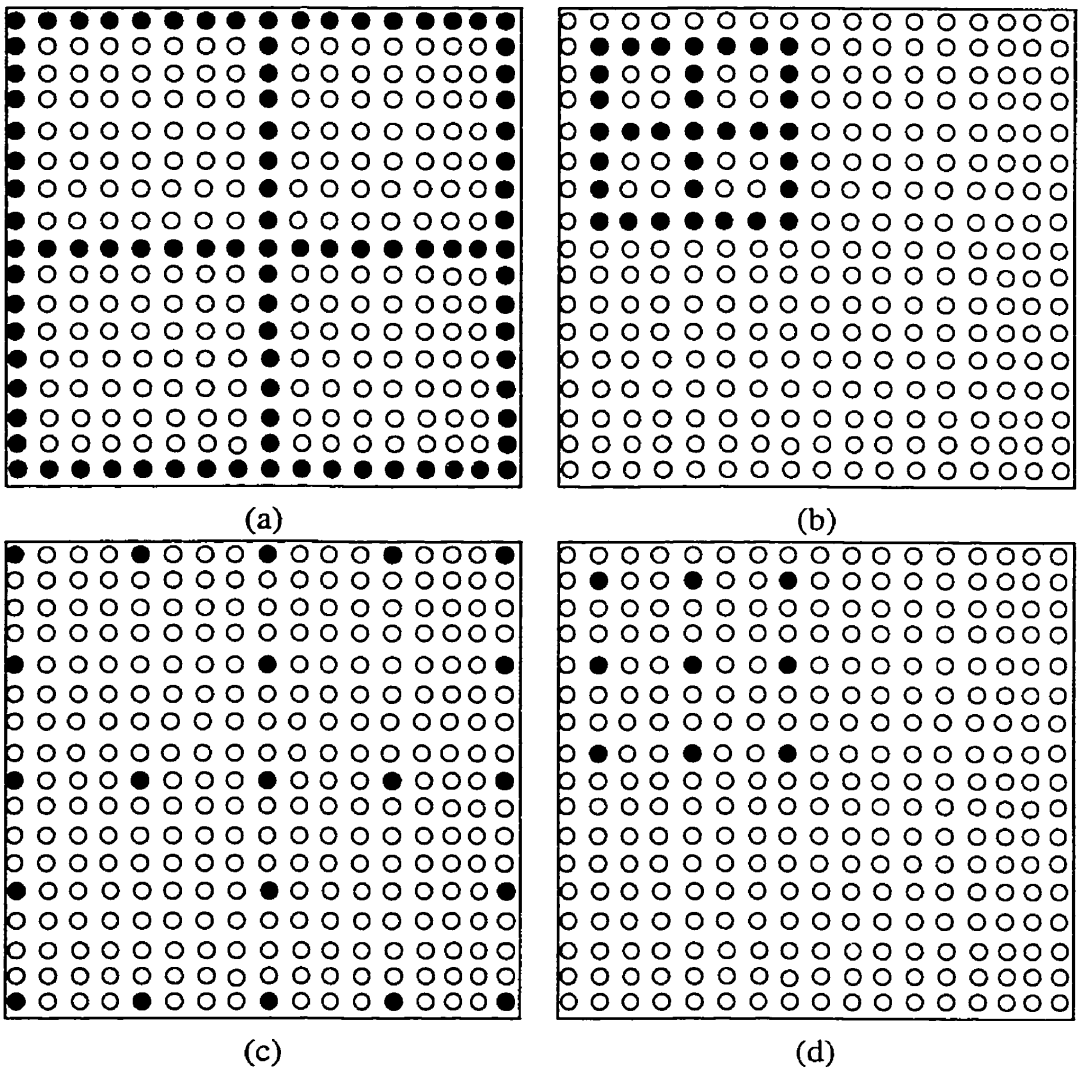


Figure 3.5: Boundary elements (•) that are kept as multiscale states $x(s)$ for 17×17 MRF in case of *non-redundant model*. Left (a) shows boundary pixels for the root node $x(0)$ and right (b) shows boundary pixels for the next level for the left-north quadrant in the quadtree $x(s\alpha_1)$ (c) shows one choice of boundary pixels for the root node $x(0)$ for the *non-redundant reduced-order model* (d) shows boundary pixels for the next level for the left-north quadrant in the quadtree $x(s\alpha_1)$.

high computational demands and numerical problems other approximate alternatives exist [50]:

- Subsample boundary pixels to a fixed percent of the number of boundary elements. This model is refered to as a *reduced order multiscale model*. This model is based on the fact that most random fields have the characteristic that neighbour elements are highly correlated. Hence, o nly a subset of those elements are needed to capture the correlation information. In this case, the sampling matrix $W(s)$ defined in (3.22) samples every K^{th} element of the finest scale process x^M along the boundaries of the children of node s [72]. Figure 3.5 (c,d) show the subsampled boundary elements for level 0 and level 1.

- Densely sample boundary elements in regions containing measurements and reduce sampling in regions far away from measurements [50].

- Coordinate transformation of the boundary pixels using wavelet transforms and keep some of the resulted coefficients [66].

- Take averages or weighted averages of the boundary elements [32].

Depending on the amount of approximation, the multiscale model fidelity is decreased. Consequently, the obtained estimates and the associated error variances become approximate. In addition, noisy artifacts appear in the obtained estimates depending on the degree of the achieved decorrelation. The advantages of reduced order or approximate models are

- Less redundant information which reduce numerical errors due to singularities

- Much lower computational and storage demands

Multiscale modeling based on sampling the boundary elements in order to satisfy the Markovianity property becomes exact and efficient when applied to Markov random processes. For general Gaussian random processes, exact multiscale realization is achieved

based on *canonical correlation realization* CCR [53]. The basic idea is to have state variables $x(s)$ at any tree node s that decorrelate all the $q+1$ subtrees connected to that node. Algorithm details can be found in [40, 50]. The CCR is based on applying singular value decomposition for each node on the tree which is computationally demanding. The complexity for CCR is $\mathcal{O}(n^4)$ where $n = N \times N$ and N is the dimension of one edge of the 2-D domain [50].

Another class of random processes that have been modeled by the multiscale framework is the $1/f$ *processes* [37]. This class of processes appears in many oceanography problems [103]. As an example, $1/f$ processes are suitable to model the surface height of the ocean [37]. The $1/f$ multiscale model where the tree states are scalar is given by

$$x(s) = x(s\tilde{\gamma}) + B_o 2^{(1-u)m(s)/2} w(s) \tag{3.25}$$

where $m(s)$ is the scale of node s and B_o is the magnitude of the process noise variance and u is some parameter in the range $0 \leq u \leq 2$ controlling the steepness of the process power spectra [103].

In this simple case, the states $x(s)$ are scalar. The state at any node is equal to its parent node (i.e., $A(s) = 1$) plus a white noise process. The white noise process is scaled exponentially as a function of scale.

From (3.25) we can observe that by moving down to finer scales, details are added as process noise $w(s)$. The state at the root node can be considered as the aggregate average over the entire process.

One advantage of $1/f$ multiscale model is that it requires low state dimension implying that large 2D estimation problems can be solved very fast.

3.5 Previous work on large-scale multiscale-based static estimation

The multiscale algorithm is mainly developed to solve static estimation problems. Consequently, in most of the previous applications of the multiscale algorithm, even though the underlying physical systems are dynamic, the estimation problems were treated as static estimation problems by considering one snapshot of the system at a time.

$$\hat{x}(t) = E(x(t)|y(k), k = t] \tag{3.26}$$

One can categorize the previous multiscale researches into three major categories:

1. *theoretical developments* [17, 40, 53, 52, 66, 67].
2. *applications to remote sensing static estimation problems* [37, 39, 72].
3. *applications to image processing problems* [38, 66, 88].

As both static estimation and remote sensing previous developments are relevant to this work, following is a brief overview of those applications.

Multiscale estimation was used in mapping ocean surface height from satellite measurements [72]. Gridding maps with $25,000$ grid points were generated in less than a minute on the current generation of workstations. This success lies behind the fact that the multiscale framework is directly suited to capturing phenomena that display hierarchical statistical structure such as stochastic processes with $1/f$ power spectra. Many natural phenomena such as the ocean surface height are modeled as $1/f$-processes. The multiscale model for the ocean height problem was given by

$$x(s) = x(s\bar{\gamma}) + (35)2^{-m(s)/2}w(s) \tag{3.27}$$

where $x(s)$ is a scalar state representing the ocean height at a particular scale and position corresponding to node s. The main results of this work are

- Successfully applying multiscale framework in solving a large-scale remote sensing static estimation problem.

- One major drawback of $1/f$ models with small tree state dimensions is the resulting artifacts in the estimates due to the inability of the model to capture the correlation structure of the process exactly and the sensitivity of the model to the spatial positioning of the multiscale tree. Two approaches were proposed for resolving this issue: averaging estimates produced by various trees at different spatial positions and incorporating the overlapping tree approach [52].

- The multiscale $1/f$ can be tailored to compensate for non-stationaries in the process prior by increasing the driving noise values $B(s)$ on those multiscale tree nodes which overlap with regions that have non-stationary statistics.

Another significant application of the multiscale algorithm in remote sensing was in mapping ocean surface temperature [34] based on hydrographic data sets. In this application, the model given in (3.22) was used to model the process. The state $x(s)$ at each node s equals a subset of the process along the boundaries of the children of s. The sampling density was controlled by a parameter proportional to the correlation length of the random process defined by its prior covariance P_x. The multiscale model parameters $(A(s), B(s))$ were then computed using (3.23),(3.24). The main reported outcomes in this work are:

- A reduced-order model was used to model the 2-D MRF process. By increasing the sampling density of the boundary elements, a higher statistical fidelity of the resulting estimates and the error statistics were obtained. However, numerical rounding errors and computational demands are increased. Decreasing the sampling den-

sity increases the algorithm computational efficiency but reduces the quality of the obtained estimates. In addition, artifacts appear in the obtained estimates.

- The multiscale model was built based on assuming a stationary model for the underlying random field. This is one of the major challenges associated with using the multiscale estimator to solve large-scale static estimation problems. Dealing with this issue is a major part of this thesis work.

In image processing area, three main applications of the multiscale framework have been investigated: Estimating optical flow [66], the surface reconstruction problem [38], and the image segmentation problem [88].

From the previous efforts and developments that were based on the multiscale framework, following is a summary of the reported features and the remaining active challenges:

Features of the multiscale estimation

- The algorithm is capable of dealing with estimation problems that involve large data sets with highly efficient computational and storage demands. This is mainly because it does not require brute-force inversion of large covariance matrices.

- The computed estimates are based on all available measurements everywhere in the field. This is in contrast to local methods that compute estimates based on local measurements and hence ignores the long correlation information that might exists in the prior.

- The algorithm provides estimation error variances at no additional computational costs. Estimation error variances provide a measure to assess the quality of the obtained estimates. In addition, error variances are necessary components in dynamic estimation problems as they are part of the prior for the next time step. One of the contributions of this work is to employ multiscale estimator in large-scale

dynamic estimation problems. Therefore, estimation error variances are essential components in the proposed method.

- Maximum-likelihood calculations and sample-path generation are two important tools in estimation and statistical analysis which can be computed efficiently by the multiscale estimator.

Active challenges with the multiscale estimation framework

- Estimation efficiency is highly dependent on the multiscale realization of the underlying process statistics. For 2-D MRF, multiscale models that can capture the statistical behaviour of the field exactly require tree state dimensions that grow with the field dimensions. This increases the fidelity of the obtained estimates and the estimation error variances. On the other hand, large state dimensions will usually result in increasing the computational demands of the estimator. In addition it might lead to numerical errors.

- Reduced-order models are used as alternatives for large-scale estimation problems. In this case the multiscale model is approximate. State-reduction increases the computational efficiency and decreases the possibility of numerical errors. However, the approximation in the obtained estimates highly increases, in addition visual artifacts appear in the estimated field.

- $1/f$ multiscale models provide fast estimation as the state dimensions are very small (scalar in most cases). These models represent good alternatives for many natural processes whose statistics are not well-defined and for large-scale problems with non-stationary priors that can't be given explicitly in a covariance matrix form due to storage issues. Building such models is totally affected by the spatial positioning of the multiscale tree. In consequence, the overlapping regions approach or averaging estimates obtained based on various tree positions should be used. This leads to more computational demands.

- In most of the previous works that dealt with large-scale static estimation problems, the assumed prior for the underlying process is considered to be stationary. However, for the following two problems, the stationarity assumption might lead to statistical inconsistency and therefore, large reduction in the estimator fidelity:
  - For large-scale 2-D MRF where a well-structured non-stationary prior is explicitly specified.
  - In dynamic estimation problems, the measurement update step transforms a stationary prior into a non-stationary one.

  In previous works, the non-stationarity in the prior was marginally tackled only in the case of $1/f$ multiscale models. Addressing the above issue for static and dynamic estimation of 2-D MRF processes is part of this thesis contribution.

3.6 Multiscale-based large-scale dynamic estimation

For large-scale 2-D dynamic estimation problems, one needs to deal with the computational and storage demands for the two main steps of the Kalman filter: the measurement update step and the dynamic prediction step. The measurements update step in the Kalman filter is composed of two components: the predicted estimates and the estimate of the prediction error based on the innovation sequence $(y(t) - C(t)\hat{x}(t|t-1))$

$$\hat{x}(t|t) = \hat{x}(t|t-1) + \hat{\tilde{e}}(t|t-1) \tag{3.28}$$

where the estimated prediction error component $\hat{\tilde{e}}(t|t-1)$ is computed by

$$\hat{\tilde{e}}(t|t-1) = K(t)(\mathrm{y}(t) - \mathrm{C}\hat{x}(t|t-1)) \tag{3.29}$$

where $K(t)$ is LSE gain and $y(t)$ is the current measurement.

By observing that estimating the prediction error is just a static estimation problem, solving this static estimation problem can be done efficiently using the multiscale estimator. The outputs of the multiscale estimator are the updated state estimates $\hat{x}(s,t|t)$, the estimation error variances $\tilde{\mathbf{p}}(s,t|t)$, and a multiscale model for the updated estimation $\tilde{P}(s,t|t)$ error process from which any cross-covariance element can be computed.

The accuracy and computational complexity of the multiscale update step are dependent on the choice of the multiscale model. In the static estimation problems, three main multiscale models have been investigated and applied (see Section 3.4):

- Multiscale models for MRFs using the subsampling approach to satisfy the Markovianity property on the tree and its model order variations.
- General models for Gaussian random fields GRFs using singular value decomposition.
- Models for processes with $1/f$ power spectra.

In the dynamic estimation context, the estimation error process at any time step is to be modeled on the tree. The following technical issues arise when using the multiscale estimator in dynamic estimation problems:

1. Which multiscale model of the above listed is more appropriate to capture the statistical structure of the error process
2. How to temporally propagate the updated estimation error variance

For the first point, selecting the appropriate model depends on the size of the problem and the desired accuracy. This issue has been investigated in [32, 50]. The difficulty and the challenging part are in the error prediction step.

$$\tilde{P}(t+1|t) = A\tilde{P}(t|t)A^T + Q \tag{3.30}$$

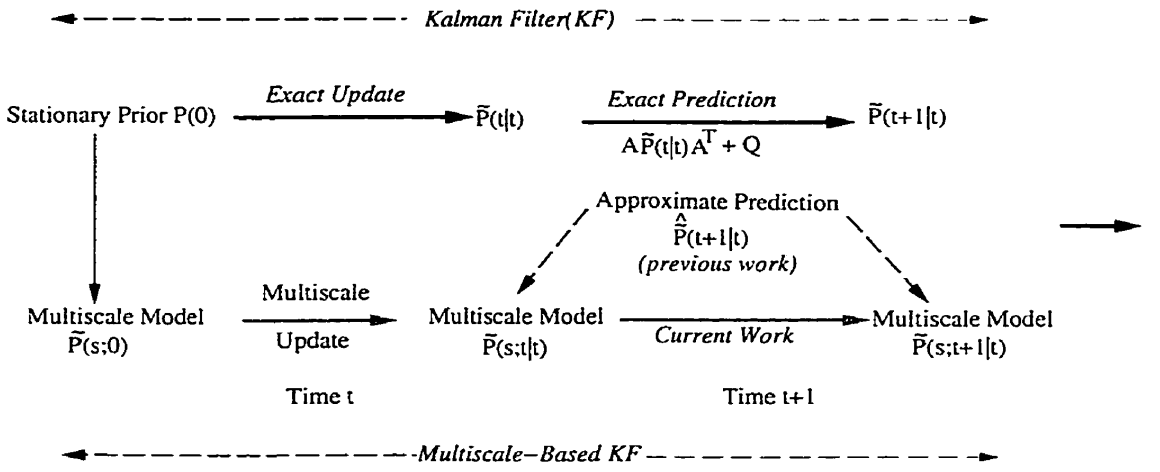


Figure 3.6: Illustration of Multiscale-based dynamic estimation with various alternatives for prediction step. The top path represents the conventional Kalman filter, the path denoted by dark dashed arrows represents the previous approach for large-scale dynamic estimation, and the bottom path represents the multiscale-based Kalman filter developed in this work.

In order to complete one step in the Kalman filter, prediction is necessary for both the updated states $\hat{x}(t|t)$ and the associated updated error covariance $\tilde{P}(t|t)$. The complexity of the standard state prediction step is $\mathcal{O}(n)$, where $n = N \times N$ and N the length of one edge of the 2-D field, which is achievable even for large-scale problems as in many cases one can have an implicit representation for the dynamics. As an example, in case the system dynamics matrix A represents a discretized diffusion process, then A can be represented by a convolutional mask and the state prediction (2.28) can be implemented by convolution instead of direct matrix-vector multiplication which requires storing A. Exact error prediction via the Lyapunov equation (2.29) requires $\mathcal{O}(n^3)$ computational and storage complexity which is totally impossible for large-scale problems. The advantage of employing the multiscale estimator is that the estimation error covariances are available but not as ordinary huge covariance matrices but in forms of multiscale error

models. For large-scale static estimation problems, diagonal elements of the estimation error model computed by the multiscale estimator are enough to assess the estimation results. However, for dynamic estimation problems, the multiscale error models need to be propagated overtime. Figure 3.6 illustrates the recursive estimator that employs the multiscale estimator. The main contribution of this work is to propose a new approximate prediction step that suits large-scale dynamic estimation problems (the gray dashed arrow in the figure). Before presenting the proposed method, the following section gives a brief summary of the previous effort on multiscale prediction step.

3.6.1 Previous formulation of Multiscale prediction step

Previous work [50] on multiscale-based dynamic estimation was mainly applied to diffusion dynamic models. The multiscale update step produces estimates for states and the associated estimation error variances. In addition, a multiscale model for the statistical structure of the updated estimation error process $\tilde{P}(s,t|t)$ is also computed. The multiscale model is completely defined by the model parameters of (3.4) which are

$$A(s,t|t) \quad B(s,t|t) \quad P(0,t|t)$$

where $A(s,t|t)$ and $B(s,t|t)$ refer to the multiscale model for the updated error process at time t and $\tilde{P}(0,t|t)$ is the covariance at the root node of the tree. Given these parameters, any off-diagonal element of the realized updated error covariance matrix $\tilde{P}(t|t)$ can be computed.

Recall that in order to propagate the estimation error covariance overtime via the Lyapunov equation (2.29), the full updated error covariance matrix $\tilde{P}(t|t)$ must be available. Eventhough the availability of multiscale model for the updated error allows the computation of the full updated error covariance, it becomes totally infeasible for large-scale problems because of both the high computational demands and the storage requirements. Implicit prediction which propagates multiscale error models instead of error covariances

is the alternative.

As it was illustrated in Section 3.4, the state $x(s)$ at any tree node (s) on the tree is written as a linear function of the fine-scale process (3.22). Similarly, for the estimation error process the tree states are given by

$$e(s,t|t) = W(s,t|t)e^M(t|t) \tag{3.31}$$

where $e^M(t|t)$ is the realized error process at the finest scale M, and $W(s,t|t)$ is a linear functional to realize the selected multiscale model. Similarly, the predicted multiscale error model $\tilde{P}(s,t+1|t)$ is given by

$$e(s,t+1|t) = W(s,t+1|t)e^M(t+1|t) \tag{3.32}$$

where A is the system discretized temporal dynamics. Recall from Section 2.4 that the error process evolves according to

$$e(t+1|t) = Ae(t|t) + w(t) \tag{3.33}$$

Therefore, from (3.31), and (3.32) the multiscale updated error model is related to the multiscale predicted error model by

$$e(s,t+1|t) = W(s,t|t)Ae(t|t) + W(s,t|t)w(t) \tag{3.34}$$

Hence, the predicted error $\tilde{P}(s,t+1|t)$ is computed by

$$\begin{aligned} \tilde{P}(s,t+1|t) &= E[e(s,t+1|t)e^T(s,t+1|t)] \\ &= W(s,t|t)AP(s,t|t)A^TW^T(s,t|t) + W(s,t|t)QW^T(s,t|t) \end{aligned} \tag{3.35}$$

Equation (3.35) is similar to the standard Lyapunov equation of the Kalman filter (2.29) with the addition of the linear functional $W(s)$ term. The multiscale predicted error

model parameters for any tree node (s) are

$$A(s,t+1|t) \quad B(s,t+1|t) \quad P(0,t+1|t)$$

But recall from (3.18), (3.20) that in order to compute $A(s,t+1|t), B(s,t+1|t)$ we need not only the predicted error covariance of state $x(s)$, i.e., $\tilde{P}(s,t+1|t)$ but also the predicted joint-statistics of $x(s)$ and its parent $x(s\bar{\gamma})$, i.e., $\tilde{P}(s,s\bar{\gamma},t+1|t)$ in addition to the predicted parent state covariance $\tilde{P}(s\bar{\gamma})$. The predicted states $x(s)$ and $x(s\bar{\gamma})$ are given by

$$\begin{aligned} e(s,t+1|t) &= W(s,t+1|t)Ae(t|t) \\ e(s\bar{\gamma},t+1|t) &= W(s\bar{\gamma},t+1|t)Ae(t|t) \end{aligned} \tag{3.36}$$

Figuring out which update cross-covariances are involved depends on the multiscale model and the temporal dynamics i.e., the product term $W(s,t+1|t)A$. Based on the multiscale models mentioned in the previous section, there are mainly three methods to incorporate the temporal effect of the dynamics on the multiscale model as illustrated in Figure 3.7

Time-varying linear functionals $W(s,t)$ for MRF models

It has been found in [50] that the error process for diffusion processes at any time can be reasonably approximated by a Markov random field. Hence, the subsampling approach for modeling the error process on the tree can be used to satisfy the tree Markovianity condition. This means that $W(s)$ is a sampling matrix keeping as a state $x(s)$ the boundary elements of its descendants. It is obvious from (3.35) that due to the mixing effect of the dynamics, linear functionals used to model the updated error process $W(s,t|t)$ are mixed. In order to reflect this mixing effect in the multiscale model, one has to keep not only the end-point pixels as boundaries but also the two nearest-neighbours (in the case of a 1-D process) to those pixels at any tree node (s)

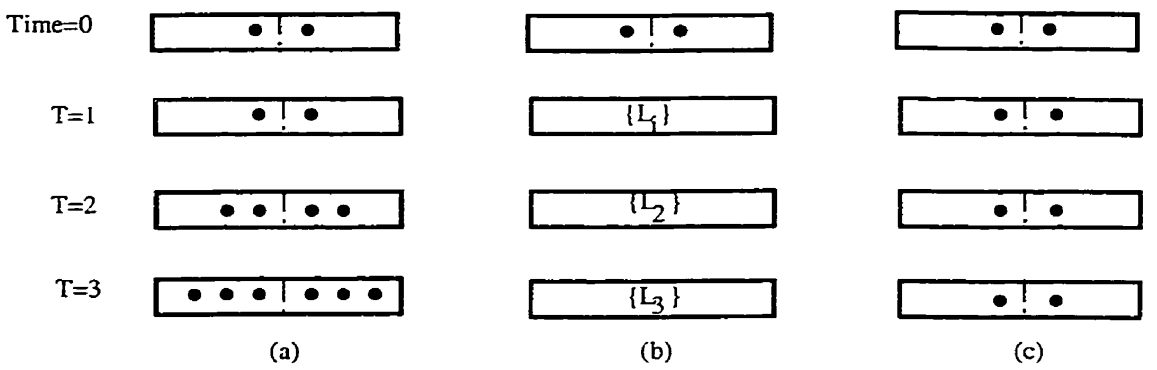


Figure 3.7: Different alternatives for incorporating the temporal dynamics effect in modeling tree states on 1-D tree. (a) illustrates the time-varying linear functionals which keep boundary elements and nearest-neighbours at anytime step in case of MRF models, (b) shows the time-varying singular values that are kept as states in case of GRF models, and (c) shows the time-invariant linear functionals which keep only boundary elements at anytime step.

and the four-nearest neighbours in the 2-D case. This is illustrated for the 1-D case in Figure 3.7 (a). Observe that this approach will eventually lead to tree states with large dimensionality. Therefore, at some point of time the computational demands of this approach will be very high which makes the multiscale-based Kalman filter computationally inefficient.

Time-varying linear functionals $W(s,t)$ for GRF models

One approach for modeling general Gaussian random fields is to use singular value decomposition to compute the tree states. This approach gives more accurate results as it avoids the Markovianity assumption for the error process. However, it is suitable only for small size problems due to the high computational demands required to compute singular values. This approach is illustrated in Figure 3.7 (b).

Time-invariant linear functionals $W(s)$

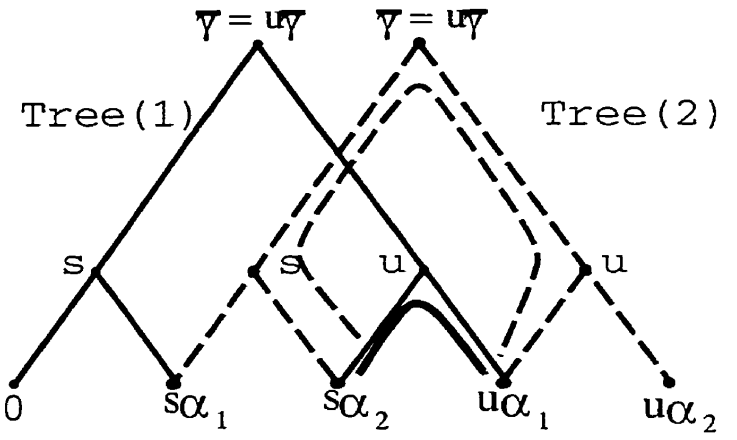


Figure 3.8: Illustration of modeling a 1-D process using multiple trees with shifted origins in order to reduce the computational effort needed to compute updated joint-statistics necessary to compute the predicted multiscale model.

In this approach, the linear functionals $W(s)$ based on the above two methods are kept constant for all time steps. This is the approach that was adopted in [50]. For time-varying systems, fixing the linear functionals implies that only approximate estimates are obtained. This approach is more suitable for implementing a large-scale steady-state filter. This approach is illustrated in Figure 3.7 (c).

3.6.2 Approaches for computing updated cross-covariances

As mentioned in the previous section, in order to compute multiscale model parameters for the predicted error process, some cross-covariances of the updated error process should be computed. The computational demands for the multiscale-based Kalman filter increase by the number of the required cross-covariances. This makes the brute-force approach (i.e., computing all the necessary cross-covariances) highly computationally inefficient. In the following, other alternatives to the brute-force approach to computing the necessary updated off-diagonal statistics are presented:

- Using multiple trees

This approach is based on using multiple multiscale models with shifted origins [32, 50]. The basic idea is that some tree states that belong to different parents on one tree may become siblings in another tree with a shifted origin. This will highly reduce the computational effort in computing their cross-covariances. The idea is illustrated in Figure 3.8 for a 1-D random process. Observe that computing the cross-covariance between states $x(s\alpha_2)$ and $x(s\alpha_3)$ on tree(2) (the tree with dashed line), one needs to compute $A(s\alpha_2), A(s), A(u), A(u\alpha_3)$ where only $A(s\alpha_2), A(u\alpha_3)$ are needed if tree(1) is used.

This approach reduces the computational demands for computing the joint-statistics between any two tree nodes, but for large-scale problems it becomes inefficient as the number of multiscale trees increases.

- Using maximum entropy

  The maximum entropy approach to complete covariance matrix has been studied in [20, 40, 64, 79]. The basic problem is given some entries of the covariance matrix, obtain estimates for the remaining entries which will produce maximum entropy of this covariance. Applying maximum entropy to dynamic estimation problems has been studied in [50]. Instead of computing all the necessary cross-covariances by brute-force, some entries are computed exactly and others are estimated based on the maximum entropy completion method.

  The main drawback of this method is that it is only applicable to 1-D processes. It is not clear how this method can be extended to 2-D.

- Using parameterized models for the error covariances

  This approach is the one that we are investigating in this work. Having parameterized error models will allow us to deal with large-scale estimation problems since any covariance entry can be computed directly based on the model. The challenge to this approach is that while the process covariance structure is stationary, the

error process covariance is not. Theoretically, there is no way to have a direct formulation for a non-stationary model that can be fitted to the process statistics. Stationarity assumptions can lead to a huge approximation as one is treating the interactions between domain pixels equally, which is not necessarily true for many cases. In this work, we investigate a new approach that can deal with these issues for large-scale 2-D problems. This approach is detailed in the next chapter.

Chapter 4

Large-Scale Dynamic Estimation: Prediction Step

In the previous chapter, we introduced the multiscale estimator and its usage in both large scale static and dynamic estimation problems. This chapter will present our alternative approach for large scale 2D dynamic estimation which is based on parameterized error models. The main subject of this chapter is the error prediction step. The measurement update step will be covered in the next chapter.

4.1 Kalman filter revisited

Recall from Chapter 2 that the Kalman filter consists of two main steps: a prediction step and an update step. The prediction step for both the state estimate $\hat{x}$ and the associated estimation error covariance $\tilde{P}$ is given by

$$\hat{\mathrm{x}}(t|t-1) = \mathrm{A}\hat{\mathrm{x}}(t-1|t-1) \tag{4.1}$$

$$\tilde{P}(t|t-1) = A\tilde{P}(t-1|t-1)\mathrm{A}^T + \mathrm{BB}^T \tag{4.2}$$

In the prediction step, the filter computes estimates of the state $x(t)$ based on the model and the previous updated estimates $\hat{x}(t-1|t-1)$. Both the deterministic and stochastic dynamics represented by the model parameters A, B are involved in the prediction step to obtain best estimates $\hat{x}(t|t-1)$ and the associated estimation error covariance $\tilde{P}(t|t-1)$.

In the update step, the predicted estimates are combined with the new available measurements at time (t) to obtain estimates based on all the data available up to time (t):

$$\mathrm{K}(t) = \tilde{P}(t|t-1)\mathrm{C}^T(\mathrm{C}\tilde{P}(t|t-1)\mathrm{C}^T + \mathrm{R})^{-1} \tag{4.3}$$

$$\tilde{P}(t|t) = \tilde{P}(t|t-1) - \mathrm{K}(t)\mathrm{C}\tilde{P}(t|t-1) \tag{4.4}$$

$$\hat{\mathrm{x}}(t|t) = \hat{\mathrm{x}}(t|t-1) + \mathrm{K}(t)(\mathrm{y}(t) - \mathrm{C}\hat{\mathrm{x}}(t|t-1)) \tag{4.5}$$

The Kalman filter deals with estimation error covariances $\tilde{P}(t|t), \tilde{P}(t+1|t)$ as matrices. For large-scale problems, it becomes impractical, if not even impossible, to compute and store such matrices. This chapter presents an approach for the prediction step which is based on using parameterized models for the error covariances with focus on 2D diffusion processes. Before presenting the approach, I begin by introducing parameterized covariance models in Section 4.1.1 and then review 2D diffusion processes in Section 4.1.2.

4.1.1 2-D stationary priors

A random process $\underline{x}$ is called wide sense stationary if its statistics satisfy the following two conditions:

1 The process has a time invariant mean, i.e.,

$$E[\underline{x}(t)] = \underline{m_x} \tag{4.6}$$

2 The correlation between any two process elements is independent of their actual spatial position, i.e.,

$$E[\underline{x}(k)\underline{x}(l)] = E[\underline{x}(k)\underline{x}(k+|k-l|)] = R(\tau_x) \tag{4.7}$$

$$\tau_x = |k-l| \tag{4.8}$$

where τ_x is called the lag, and $R(\tau_x)$ is the process autocorrelation at lag τ_x.

The same concept is extended for 2D random fields, where the correlation between any two field elements is given by

$$E[x(i,j)x(i+\tau_i, j+\tau_j)] = E[x(0,0)x(\tau_i,\tau_j)] \tag{4.9}$$

where τ_i, τ_j are the horizontal and vertical lags, respectively

Stationary prior models for image and signal processing applications have been studied in various contexts and applied widely to several applications related to the field: motion estimation and image coding [58] and image modeling and restoration [47, 55], to name a few. In general, Gaussian or exponential functions are used as priors to model natural phenomena. This is usually an intuitive model since the relationship between any two random field elements decays as a function of the distance between them. There are two choices of exponential 2D priors:

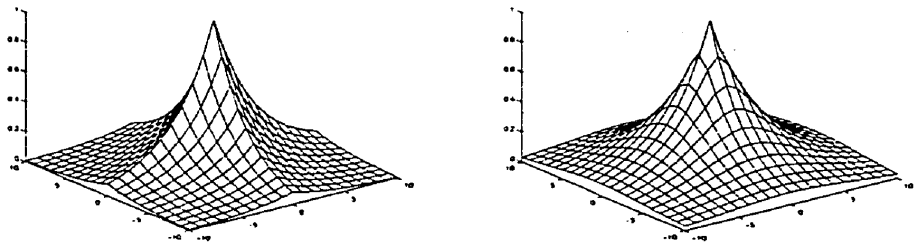
1 A separable exponential prior model given by

$$P_{\tau_x,\tau_y} = \sigma^2 e^{-\{\frac{|\tau_x|}{l_x}+\frac{|\tau_y|}{l_y}\}} \tag{4.10}$$

This correlation model is depicted in Figure 4.1 (a).

2 A non-separable exponential correlation coefficient model given by

$$P_{\tau_x,\tau_y} = \sigma^2 e^{-\{\frac{\tau_x^2}{l_x^2}+\frac{\tau_y^2}{l_y^2}\}^{\frac{1}{2}}} \tag{4.11}$$



(a) (b)

Figure 4.1: Illustration of two 2D exponential correlation functions. (a) is separable, (b) is non-separable.

This correlation model is depicted in figure Figure 4.1 (b).

In the above two exponential models, σ^2 is the overall domain variance, and τ_x and τ_y represent the lags in the x and y directions. l_x, l_y represent the correlation lengths (the distance between two field elements to have a correlation coefficient value equal to $\frac{1}{e}$) in both directions where larger values indicate higher correlation between the process elements and hence, a smoother process structure. Figure 4.2 shows that varying the correlation length in a model controls the amount of interaction between the field elements. Note that for an isotropic (homogeneous) correlation structure, the correlation length is the same in both the horizontal and vertical directions, i.e., $l_x = l_y$. Observe from Figure 4.1 (a), however, that for the case of the separable prior (4.10), it is not exactly isotropic even when $l_x = l_y$. In this thesis, we will use the word isotropic to mean a prior that has equal correlation length in both horizontal and vertical directions. The above exponential priors handle only positively correlated random variables which is enough for the correlation structures of diffusion processes .

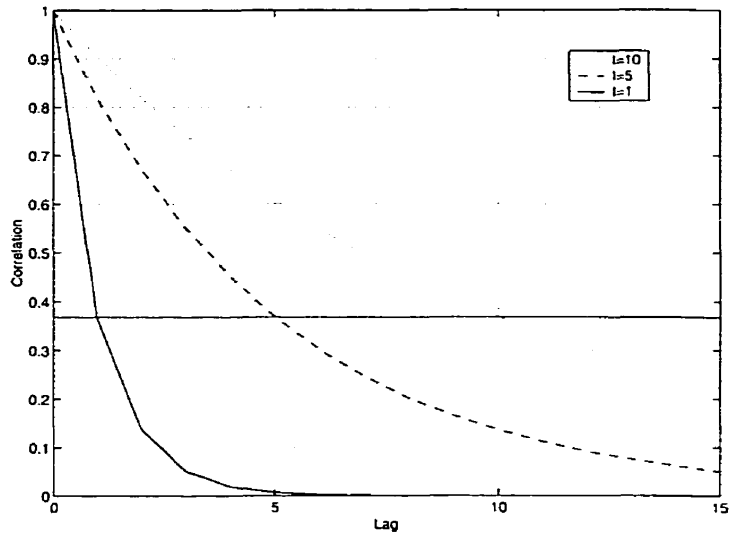


Figure 4.2: Illustration of the effect of the correlation length on the process correlation structure. Observe that increasing the correlation length implies a higher correlation between the process elements.

4.1.2 Review of Diffusion Dynamics

In this research, we focus our study on 2D diffusion processes. Therefore, the dynamic operator A represents the discretized 2D diffusion process. In this section we present a brief review of such processes. The diffusion process is a well understood process with a simple behavior. It has been used for several applications such as image restoration [56], and in estimation problems [29]. In addition, the diffusion model is a reasonable model for the ocean surface temperature problem that we will apply our method to.

For a 1D process, the diffusion process is governed by the following stochastic partial differential equation (PDE) [31, 98]

$$\frac{\partial T(x,t)}{\partial t} = a \cdot \frac{\partial^2 T(x,t)}{\partial x^2} - b \cdot T(x,t) + \gamma \cdot w(x,t) \tag{4.12}$$

where $T(x,t)$ is the temperature distribution at spatial position x and time t, a is the diffusion parameter which is media dependent, b is a heat loss term representing the heat

transferred to the surroundings, and $w(x,t)$ is Gaussian white noise with variance γ^2.

For a 2D process, the diffusion process is given by the following

$$\begin{aligned}\frac{\partial T(x,y,t)}{\partial t} &= a_x \cdot \frac{\partial^2 T(x,y,t)}{\partial x^2} + a_y \cdot \frac{\partial^2 T(x,y,t)}{\partial y^2} - b \cdot T(x,y,t) \\ &+ \gamma \cdot w(x,y,t) \end{aligned} \tag{4.13}$$

where $T(x,y,t)$ is the temperature at position (x,y) at time t, and $w(x,y,t)$ is Gaussian white noise with unit variance.

There exist various finite-difference schemes [31, 98] to discretize (4.13) in order to construct a system of difference equations

$$\underline{x}(t+1) = A\underline{x}(t) + w(t) \tag{4.14}$$

where $\underline{x}$ is a vector representing the 2D process ordered lexicographically. Using a Forward-Euler discretization scheme [31, 98] A is penta-diagonal.

Matrix A can be represented implicitly by the following convolutional molecule C_m

$$C_m = \begin{bmatrix} 0 & \beta & 0 \\ \beta & \alpha & \beta \\ 0 & \beta & 0 \end{bmatrix} \tag{4.15}$$

Where

$$\alpha = 1 - 4 \cdot a \cdot \frac{\Delta t}{(\Delta x)^2} - b \cdot \Delta t \text{ and} \tag{4.16}$$

$$\beta = a \cdot \frac{\Delta t}{(\Delta x)^2} \tag{4.17}$$

assuming an equal spatial discretization interval in both x, y directions. So we can alter-

natively express (4.14) as

$$x(t+1) = C_m * x(t) + w(t) \tag{4.18}$$

where $(*)$ stands for convolution operation and all variables are in their actual 2D forms, i.e., not in vector forms.

Our knowledge of the process dynamic matrix A will allow us to study the process statistics at steady-state by solving the Lyapunov equation

$$P_x = AP_xA^T + Q \tag{4.19}$$

It is worth mentioning that the diffusion parameters affect the correlation structure of the process in steady-state. By modifying the diffusion parameters α, β, one can control the amount of correlation between the process elements.

4.2 Large-scale approximate prediction step

4.2.1 State prediction

In general, exact state prediction according to (4.1) is achievable. In many cases, one can exploit the sparsity of the dynamics operator A to overcome the storage issue when dealing with large-scale systems. For diffusion processes, A can be represented implicitly by the kernel given in (4.15) and, as a result, exact state prediction can be done efficiently by convolving the updated estimates $\hat{x}(t|t)$ with the kernel C_m

$$\hat{x}(t+1|t) = C_m * \hat{x}(t|t) \tag{4.20}$$

4.2.2 Error prediction

Recall that the estimation error is propagated through time by

$$\tilde{P}(t+1|t) = A\tilde{P}(t|t)A^T + Q \tag{4.21}$$

For large-scale dynamic estimation problems, exact error prediction by (4.21) is impossible. From a storage perspective, both the temporal dynamics A and the updated estimation error $\tilde{P}(t|t)$ can not be represented explicitly. In addition, from a computational perspective, matrix multiplication is of order $\mathcal{O}(n^3)$, where n is the 2D domain size, which is infeasible for large 2-D dynamic estimation problems.

Although, in some cases, by exploiting the sparsity of A the matrix-matrix multiplication can be done efficiently, the storage problem of the resulting irregular and non-stationary structure of $\tilde{P}(t+1|t)$ is still a problem.

Our approach for the error prediction step is to parameterize the error covariances $\tilde{P}(t|t), \tilde{P}(t+1|t)$.

In fact, a covariance matrix P can be expressed in terms of

1. The main diagonal elements $\text{cov}(i,i)$ which are the error variances, given by

$$\text{diag}(P) = \mathbf{p} = \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \vdots \\ \sigma_n^2 \end{bmatrix} \tag{4.22}$$

2. The off-diagonals elements or cross-covariances $\text{cov}(i,j)$ expressed in terms of standard deviations and correlation coefficients ρ

$$\text{cov}(i,j) = \sigma_i \sigma_j \rho_{ij} \tag{4.23}$$

where $-1 \leq \rho_{ij} \leq 1$

Equivalently, we can express the covariance matrix P in a matrix form, as

$$\begin{aligned} P &= \mathbf{p}\mathbf{p}^T \odot \Phi \\ &= \begin{bmatrix} \sigma_1\sigma_1 & \sigma_2\sigma_1 & \cdots & \sigma_n\sigma_1 \\ \sigma_1\sigma_2 & \sigma_2\sigma_2 & & \vdots \\ \vdots & & \ddots & \\ \sigma_1\sigma_n & \cdots & & \sigma_n\sigma_n \end{bmatrix} \odot \begin{bmatrix} 1 & \rho_{21} & \cdots & \rho_{n1} \\ \rho_{12} & 1 & & \vdots \\ \vdots & & \ddots & \\ \rho_{1n} & \cdots & & 1 \end{bmatrix} \end{aligned} \tag{4.24}$$

where Φ is the correlation coefficient matrix and the symbol $\odot$ refers to element-by-element multiplication. So in terms of the error prediction step given by (4.21), we can express the involved estimation error covariance matrices in terms of their components

$$\{\tilde{\mathbf{p}}(t|t)\tilde{\mathbf{p}}(t|t)^T\}^{\frac{1}{2}} \odot \Phi(t|t) \overset{\text{Prediction}}{\Longrightarrow} \{\tilde{\mathbf{p}}(t+1|t)\tilde{\mathbf{p}}(t+1|t)^T\}^{\frac{1}{2}} \odot \Phi(t+1|t) \tag{4.25}$$

Observe from (4.25) that in order to compute the predicted estimation error covariance $\tilde{P}(t+1|t)$ according to (4.21), we need the two components of the updated estimation error covariance $\tilde{P}(t|t)$. In this work, we adopt the multiscale estimator to solve the update step. The multiscale estimator provides in addition to the updated state estimates $\hat{x}(t|t)$ the updated estimation error variances $\tilde{\mathbf{p}}(t|t)$ (i.e the first component), and the updated estimation error process correlation structure $\Phi(t|t)$ as a multiscale tree model $\tilde{P}(s,t|t)$ (The reader is referred to Section 3.6 for more detail).

As a fact, the correlation coefficient between any two states can be computed from the multiscale updated error model $\tilde{P}(s,t|t)$, but this becomes highly inefficient when dealing with large-scale problems, as presented in Section 3.6. Even the availability of all of the necessary updated estimation error cross-covariances does not allow exact error prediction according to (4.21) because of the infeasible storage requirements for both the temporal dynamics matrix A and the full updated estimation error covariance matrix $\tilde{P}(t|t)$.

Expressing the estimation error covariance matrices as in (4.25) helps us to develop approximate error prediction methods that can predict the two components of the estimation error efficiently. In this work, we propose using a parameterized model given by (4.10), or (4.11) for the error process correlation structure Φ which has to satisfy the following constraints:

1. The resulting error covariance matrix is both positive definite and symmetric. These are two requirements for any valid covariance matrix.

2. The correlation structure imposed by the model Φ should be consistent with the statistics of the underlying process. This can be determined by looking at the process correlation structure obtained by solving the Lyapunov equation (4.19).

For diffusion dynamics, the process correlation structure decays exponentially with correlation length depending on the diffusion dynamics parameters. Hence, we model the error correlation structures $\Phi(t|t), \Phi(t+1|t)$ at any time step by an exponential model which also satisfies the positive-definite condition.

Now, given the above approach for modeling the error process correlation structure, the error prediction step reduces to (i) inferring the predicted / updated correlation length at any element of the 2D error process and (ii) propagating the updated error variances $\tilde{\mathbf{p}}(t|t)$.

Our approach to these two requirements is presented in the following sections.

4.2.3 Inferring the predicted / updated correlation structure

The spatial correlation structure of the error process at any time step is not spatially stationary. Even if the process prior $P_x = P(0)$ is stationary, once the field is updated by irregular measurements the posterior (i.e., the prior for the next time step $\tilde{P}(t+1|t)$) becomes spatially non-stationary. The issue that we are addressing here is how to reflect such non-stationarity in the parameterized error models given by (4.10) or (4.11).

Our approach is to estimate the correlation length $L(t|t)$, $L(t+1|t)$ of the updated/predicted 2D error processes. Given a non-stationary covariance matrix P, the process correlation coefficient matrix can be computed by

$$\Phi = \frac{P}{\{pp^T\}^{\frac{1}{2}}} \tag{4.26}$$

where the division in (4.26) is *point-wise*.

Now, under the assumption that the correlation ρ between any two elements of the error process is given by the exponential model (4.10) and considering the first-order neighbours $\mathcal{N}$, the correlation length at any element $e(i,j)$ can be estimated by

$$L(e(i,j)) = \frac{1}{4}\sum_{k,l\in\mathcal{N}} -\frac{1}{\ln(\rho(e(i,j)e(k,l))} \tag{4.27}$$

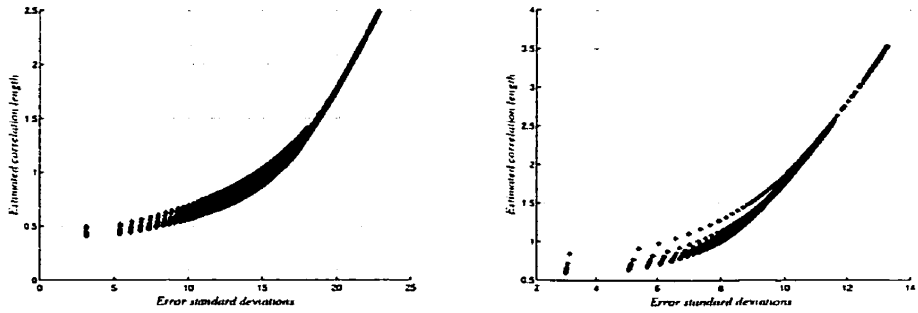
where $\rho(e(i,j)e(k,l))$ is the correlation coefficient between two neighbouring elements of the 2D process and $\mathcal{N}$ refers to the four first-order neighbours of element (i,j).

Although the above is only applicable for small size problems, it helps us construct empirical relationships between the updated / predicted estimation error variances and the corresponding updated / predicted correlation lengths.

$$\tilde{\mathbf{p}}(t|t) \implies L(t|t) \tag{4.28}$$

$$\tilde{\mathbf{p}}(t+1|t) \implies L(t+1|t) \tag{4.29}$$

where $L(t|t), L(t+1|t)$ are two dimensional matrices containing the estimated correlation length at any element of the updated and predicted 2D processes, respectively. Inferring the relationships for large-scale 2D problems is done by applying the exact Kalman filter to a similar but a smaller size problem and observing the relationship between the estimation error standard deviations and the 2D correlation length. We observed empirically that both the updated and predicted relationships take the same



(a) (b)

Figure 4.3: Scatter plots depicting the empirical relationship between error standard deviations and the corresponding correlation length for two different diffusion processes. (a) Process with correlation length (4). (b) Correlation length 12.

form. Figure 4.3 illustrates such a relationship for two different diffusion processes with steady-state correlation lengths (4) and (12). Observe that above a certain value σ_λ, of the estimation error standard deviations the relationship between the error standard deviations $\sqrt{\tilde{\mathbf{p}}}$ and the corresponding correlation lengths L can be reasonably approximated by a linear relationship $f(l)$, i.e., a straight line equation. By modeling this linearity, we will be able to extrapolate the correlation lengths for error standard deviations values beyond the empirical ones.

It is important to mention that the empirical relationship does not cover the whole range of the expected process correlation length spectrum, i.e., the minimum and maximum limits $[l_{small}\ \ l_{large}]$. The minimum and maximum limits $[l_{small}\ \ l_{large}]$ are obtained by solving both the following discrete-time Riccati and Lyapunov equations, respectively

$$\tilde{P}_p = A\tilde{P}_pA^T + Q - AK(t)C\tilde{P}_pA^T \tag{4.30}$$

$$P_x = AP_xA^T + Q \tag{4.31}$$

Let l_{min},l_{max} refer to the minimum and maximum correlation lengths realized by the empirical relationship and let σ_{min} and σ_{max} refer to their corresponding minimum and

maximum estimation error standard deviations. Also let l_{model} refer to the interpolated correlation length based on the empirical relationship for all $\sigma_{min} \leq \sigma \leq \sigma_\lambda$, then estimating the predicted/updated correlation length L_i at the i^{th} element of the 2D process is done as follows

$$L_i = \begin{cases} l_{small} & \text{if } \sigma_i < \sigma_{min} \\ l_{model} & \text{if } \sigma_{min} \leq \sigma_i \leq \sigma_\lambda \\ f(L_i) & \text{if } \sigma_i > \sigma_\lambda \end{cases} \tag{4.32}$$

4.2.4 Propagating the updated error variances

In this work, the following two models for propagating the updated error variances $\tilde{\mathbf{p}}(t|t)$ were studied.

A- **Modulating the updated error variances**

This is a simple intuitive model in which we assume that each diagonal element of the updated error covariance evolves separately (i.e., a diagonal updated error covariance matrix is assumed, $\tilde{P}(t|t) = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$). Based on this assumption, the predicted error variances $\tilde{\mathbf{p}}(t|t)$ are computed by

$$\tilde{\mathbf{p}}_\mathbf{i}(t+1|t) = \alpha^2 \tilde{\mathbf{p}}_\mathbf{i}(t|t) + \gamma_i \tag{4.33}$$

where α is the element of the main diagonal of the diffusion discrete temporal dynamics matrix A and γ_i is the process noise variance for state x_i and i refers to the i^{th} state position. Now, given $\tilde{\mathbf{p}}(t+1|t)$ and the empirical relationship between $\sqrt{\tilde{\mathbf{p}}(t+1|t)}$ and $L(t+1|t)$ as described in the previous section, any element of the rest of the prediction estimation error covariance (i.e., the cross-covariances) is given by

$$\hat{\tilde{P}}(t+1|t) = \{\tilde{\tilde{\mathbf{p}}}(t+1|t)\tilde{\tilde{\mathbf{p}}}(t+1|t)^T\}^{\frac{1}{2}} \odot \Phi(t+1|t) \tag{4.34}$$

where the predicted error correlation structure is $\Phi(t+1|t)$. Realizing a spatially non-stationary correlation structure $\Phi(t+1|t)$ based on both $\tilde{\mathbf{p}}(\mathbf{t}+\mathbf{1}|\mathbf{t})$ and $L(t+1|t)$ will be covered in the next chapter.

Although this model is computationally not demanding, it has the following disadvantages: (i) the model assumes that each estimation error pixel $\sigma^2(t|t)$ evolves independently. Therefore, the mixing effect of the diffusion dynamics is not reflected in the predicted estimation error variances, and (ii) the model clearly underestimates the predicted error variances because it ignores other terms (cross-covariances) involved in propagating the updated error variances.

B- **Exact dynamics applied to the updated error variances**

In this model discretized temporal dynamics parameters are applied exactly to the updated error variances according to (4.21). Formally this is equivalent to

$$\tilde{\mathbf{p}}(t+1|t) = \text{diag}(A\tilde{P}(t|t)A^T + Q) \tag{4.35}$$

Exact temporal prediction of the updated error variances requires figuring out what elements of the updated error covariance (variances and cross-covariances) are needed in order to compute the product $A\tilde{P}(t|t)A^T$ exactly.

Recall that for diffusion dynamics the discretized temporal dynamics matrix A can be represented implicitly by (4.15). Predicting the estimation error variance $\sigma^2_{\varepsilon(i,j)}(t+1|t)$ at any state requires applying (4.15) to the error field in a way such that (4.21) is satisfied. This requires that for a given error pixel, the updated error variances and the joint statistics of the first-order neighbourhood pixels depicted in Figure 4.4 be modulated by the temporal dynamics values given in (4.15). Based on the above, we find that the expression for predicting the estimation error variance at

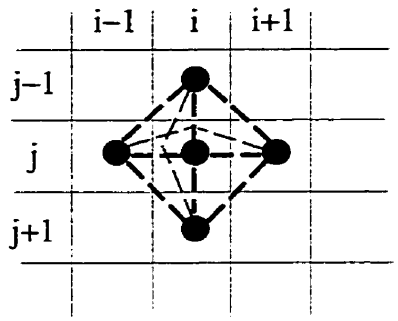


Figure 4.4: Part of the 2D domain of the updated estimation error variances $\tilde{\mathbf{p}}(\mathbf{t}|\mathbf{t})$ which illustrates the interactions (cross-covariances) between the middle pixel at the (i,j) position and its four first-order neighbours that need to be computed in order to compute the predicted estimation error variance for the (i,j) pixel

any field state is given by

$$\begin{aligned}
\tilde{\mathbf{P}}_{x(i,j)(t+1|t)} &= \alpha^2\sigma^2_{x(i,j)}(t|t) + \beta^2 \sum_{\substack{(k\in\{-1,1\},c=0);\\(k=0,c\in\{-1,1\})}} \sigma^2_{x(i+k,j+c)}(t|t) \\
&+ 2\alpha\beta \sum_{\substack{(k\in\{-1,1\},c=0);\\(k=0,c\in\{-1,1\})}} \mathrm{cov}\{x(i,j)x(i+k,j+c)\}(t|t) \\
&+ 2\beta^2\mathrm{cov}\{x(i+1,j)x(i,j-1)\}(t|t) + 2\beta^2\mathrm{cov}\{x(i+1,j)x(i,j+1)\}(t|t) \\
&+ 2\beta^2\mathrm{cov}\{x(i,j-1)x(i-1,j)\}(t|t) + 2\beta^2\mathrm{cov}\{x(i,j+1)x(i-1,j)\}(t|t) \\
&+ 2\beta^2\mathrm{cov}\{x(i-1,j)x(i+1,j)\}(t|t) + 2\beta^2\mathrm{cov}\{x(i,j+1)x(i,j-1)\}(t|t) \\
&+ \gamma_{ij}I \qquad (4.36)
\end{aligned}$$

Observe that in order to compute the predicted diagonal elements we need to compute some of the cross-covariances of the updated error covariance $\tilde{P}(t|t)$. Recall that any off-diagonal element of the updated estimation error covariance can be written in terms of the updated error standard deviations and the updated correlation coefficient as

$$\mathrm{cov}(i,j)(t|t) = \sigma_i(t|t)\sigma_j(t|t)\rho_{ij}(t|t) \qquad (4.37)$$

where $\sigma_i(t|t), \sigma_j(t|t)$ are the updated estimation error standard deviations which are available from the update step, and $\rho_{ij}(t|t)$ is the correlation coefficient. Given our

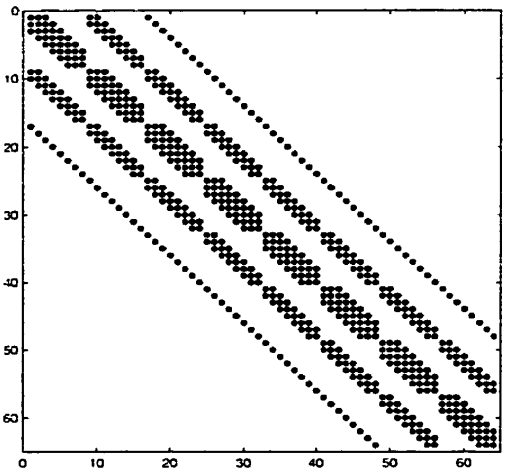


Figure 4.5: Elements of the updated estimation error covariance $\tilde{P}(t|t)$ that are needed to propagate updated error variances exactly according to (4.21) for an 8×8 domain

choice of assuming a parameterized model for the error covariances and given the empirical relationship between $\sqrt{\tilde{\mathbf{p}}(t|t)}$ and $L(t|t)$, the required correlation coefficients $\rho_{ij}(t|t)$ between any two error process elements can be computed directly from the assumed parametric correlation model $\Phi(t|t)$.

This computation is not demanding as the number of necessary cross-covariances is very small in comparison to the size of the covariance matrix. As an example, Figure 4.5 depicts the elements that are needed to propagate all of the updated error variances for a 8×8 2-D process; the total number of necessary cross-covariances is less than 17% of the 2D covariance matrix size which is 64×64 in this case. Now, given $\hat{\tilde{\mathbf{p}}}(t+1|t)$ and $L(t+1|t)$, the rest of the predicted error covariance elements are given by

$$\hat{\tilde{P}}(t+1|t) = \{\tilde{\mathbf{p}}(t+1|t)\tilde{\mathbf{p}}(t+1|t)^T\}^{\frac{1}{2}} \odot \Phi(t+1|t) \tag{4.38}$$

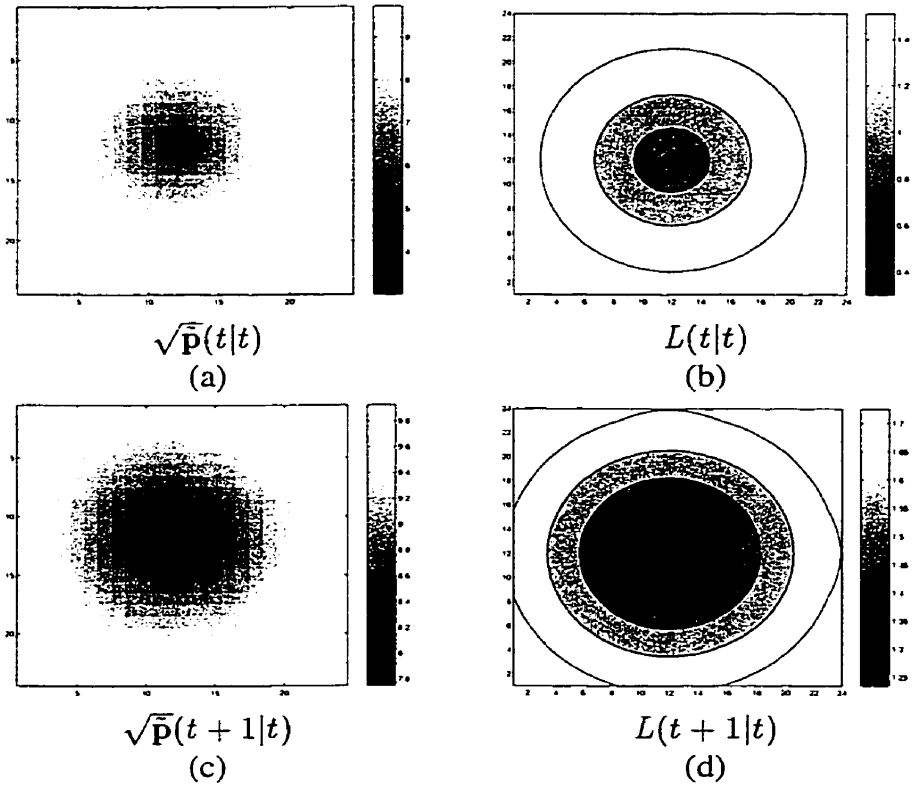


Figure 4.6: Illustration of the steps for propagating the updated error variances. (a) Updated error std. dev. $\sqrt{\tilde{\mathbf{P}}}(1|1)$. (b) Estimated correlation length $L(1|1)$ contours. (c) Predicted error std. dev. $\sqrt{\tilde{\mathbf{P}}}(20|19)$. (d) Estimated correlation length $L(20|19)$

Figure 4.6 illustrates the complete error prediction step. The right-hand-side panels show the estimated updated / predicted correlation lengths from the updated / predicted error standard deviations depicted in the left-hand-side. Observe that:

- The variations in the correlation length $L(t|t)$ levels Figure 4.6(b) after updating the field with a single measurement in the center of the domain clearly show the spatial non-stationarity of the posterior $\tilde{P}(t|t)$.
- The prediction step clearly reflects the changes in the error variances $\tilde{\mathbf{p}}(t+1|t)$ and the corresponding correlation lengths $L(t+1|t)$. This is depicted in panels

Figure 4.6(c,d) where for this case observe that there is an increase in the process correlation length at all process elements.

An important issue which was not covered in this chapter is how to realize the non-stationarity in the predicted error process correlation structure $\Phi(t+1|t)$ based on both the predicted estimation error variances $\tilde{\mathbf{p}}(\mathbf{t}+\mathbf{1}|\mathbf{t})$ and the estimated predicted correlation length $L(t+1|t)$. This will be covered in the next chapter.

Chapter 5

Large-Scale Non-stationary Static Estimation: Update step

The previous chapter covered the large-scale prediction step. In this chapter, we present a new approach for large-scale 2D non-stationary static estimation such as the Kalman filter update step. Although the stationarity assumption might be reasonable in the case of large-scale 2D static estimation problems, it leads to statistical invalidity for dynamic estimation problems where irregular spaced measurements change a spatially stationary prior into a spatially non-stationary prior. The update step in the Kalman filter (4.3),(4.4),(4.5) is just a static estimation problem. Typically the most challenging aspect of the update step is the matrix inversion in (4.3). In this work, we propose using the multiscale estimator presented in Chapter 3 which efficiently solves the update step and produces $\tilde{\mathbf{p}}(t|t)$ as needed for prediction. One difficulty when employing the multiscale estimator in dynamic estimation problems is how to provide a statistical prior which reflects the non-stationarity in the error process statistics.

Before presenting our approach for non-stationary static estimation, we begin this chapter by revisiting static estimation and then illustrate the limitation of static estimation based on stationary priors.

5.1 Large-scale static estimation revisited

Recall from Section 2.2 that optimal state estimates based on the linear least square estimation criteria are computed by

$$\hat{\underline{x}} = \underline{m_x} + P_x C^T (C P_x C^T + R)^{-1} (\underline{y} - C\underline{m_x}) \tag{5.1}$$

and the associated error covariance is

$$E[\tilde{\underline{x}}\tilde{\underline{x}}^T] = \tilde{P} = P_x - P_x C^T (C P_x C^T + R)^{-1} C P_x \tag{5.2}$$

Observe that the above two equations require the availability of the whole process prior covariance matrix P_x. For 2D processes of size $N \times N$, the covariance matrix P_x is of size $N^2 \times N^2$. If the 2D domain is large then storing P_x is impossible and the exact solution for large-scale static estimation problems by (5.1), (5.2) is infeasible. FFT methods, mentioned in Section 2.2.4, can provide approximate estimates only under special statistical restrictions. In addition, iterative methods [43, 44, 96] require the knowledge of the exact prior P_x in order to converge to the true solution given (5.1), (5.2). However storing the whole prior and also the whole posterior $\tilde{P}$ matrix is infeasible for large-scale 2D problems.

We will begin with a discussion on large-scale stationary static estimation before presenting our approach to the non-stationary case.

5.2 Limitations of large-scale 2D static estimation based on stationary prior

When computing estimates based on stationary priors, the prior covariance matrix P_x required in both (5.1),(5.2) is constructed using stationary prior models (4.10),(4.11).

It is important to mention that the stationarity assumption for the 2D random field does not lead to any reduction in the computational demands. Although the storage demands for the prior are low, storing the whole posterior $\tilde{P}$ is hard because of the non-stationarity.

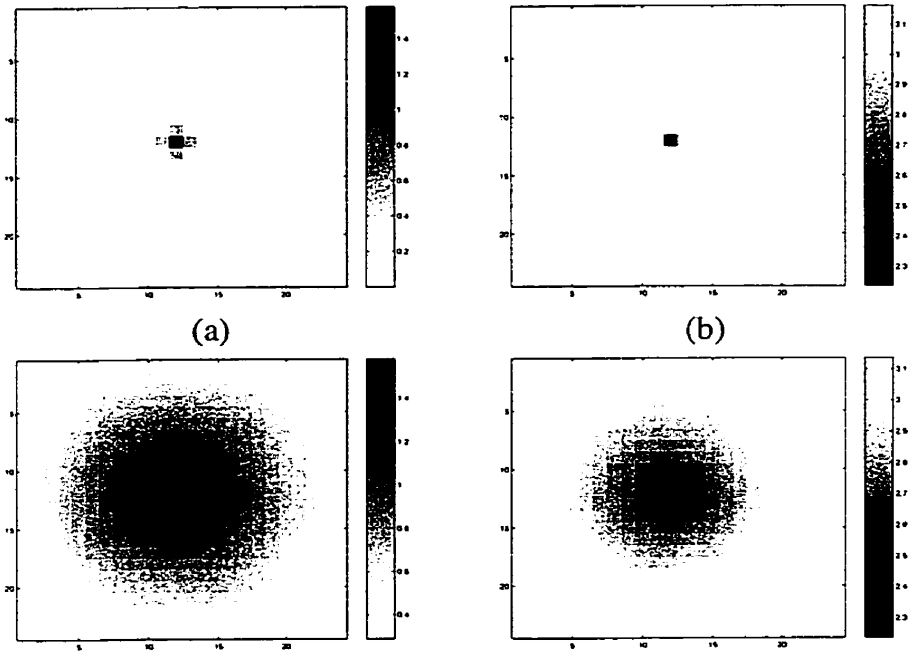
The stationarity assumption might be a reasonable assumption especially when dealing with large-scale random processes. On the other hand, for at least the following two problems, the stationarity assumption might lead to statistical invalidity and therefore, large reduction in the fidelity of the computed estimates:

The first problem is when the large-scale 2D processes has a well-structured non-stationary prior. In this case, if one assumes stationarity and underestimates the correlation length at some field regions, then the accuracy of the computed estimates at those regions will be highly affected. The effect of any measurement will be local to the measured points. On the other hand, an overestimated correlation length might mix the effect of measurements and lead to inaccurate estimates.

In addition, the associated estimation error variances will be overestimated if the correlation length is underestimated. Furthermore, if the correlation length is overestimated, then the estimation error variances will be underestimated.

As an example, Figure 5.1 shows the estimates computed for a field of size 24×24 with a single measurement at position $(12, 12)$. In Figure 5.1(a,b) estimates are computed based on an isotropic stationary prior with a short correlation length that is equal to 1. Observe that the effect of the measurement is nearly local and the estimation error variances are small only in the vicinity of the measured position and get larger as we move away from the measured position. In Figure 5.1(c,d) estimates are computed based on a stationary prior with a correlation length that is equal to 10. The effect of the measured pixel is nearly global (covers a larger area) in addition estimation error variances are small in a larger region than in the first case. This makes the selection of the correlation length in (5.1), (5.2) a very crucial issue.

The second problem is related to large-scale dynamic estimation. In this case, the



(a) (b)

(c) State estimates (d) Error standard deviations

Figure 5.1: Illustration of effect of the correlation length on the state estimates and the associated estimation error variances for a 2D process of size 24×24 and a measurement at position (12,12). Estimates in (a) are obtained based on an isotropic exponential prior with correlation length $l = 1$. An isotropic prior with correlation length $l = 10$ is used to compute estimates in (c).

measurement update step transforms a stationary prior $P_x(0)$ into a non-stationary one $\tilde{P}(0|0)$ in the case of an irregularly spaced measurement structure as depicted in Figure 4.6(d). This case is the one relevant for this thesis.

5.3 Discussion on approaches for modeling 2D non stationary priors

In the previous chapter, we showed that the predicted error model used by the proposed prediction consists of two components: the predicted error variances $\tilde{\mathbf{p}}(t+1|t)$ and the predicted correlation length at all domain elements $L(t+1|t)$. The question now is how to build a non-stationary prior model based on these components so that the updated estimates are non-stationary.

Consider a 2D field x of size $N \times N$ that has a non-stationary prior given by a covariance matrix P_x. Spatial non-stationarity implies that for any two field elements $x(i,j), x(k,l)$, their statistical relationship $E[x(i,j)x(k,l)]$ depends on their actual positions and not only on their spatial separation. For large-scale 2D domains, although the statistical structure of the whole domain is globally non-stationary, some regions of the 2D domain can be considered as locally stationary with a fixed correlation length l. In the following discussion we will use $\Gamma(l)$ to refer to a region of the 2D domain that has a correlation length equal to l. Therefore, the correlation structure of all elements $\underline{q} \in \Gamma(l)$ can be given by an appropriate stationary prior model with a correlation length l_1. Other regions in the domain can have a correlation structure with a correlation length l_2 where $l_1 \neq l_2$. The non-uniformity in the correlation length for all the domain regions is the source of the global spatial non-stationarity.

As mentioned earlier, for a large-scale 2D process, it is impossible to have a prior in a form of a matrix due to the huge storage demands. One can instead have an implicit representation for the non-stationary prior by using a stationary prior as given in (4.10) or (4.11) with *varying both the correlation length l and the variance depending on the location of any two pixels in the domain.* For simplicity, consider for example a spatially varying prior for a 1D process given by

$$P(x_i, x_k) = \sigma_i \sigma_k \exp\left\{-\frac{|i-k|}{(l(x_i)+l(x_k))/2}\right\} \tag{5.3}$$

A covariance matrix P_x constructed by (5.3) will represent a non-stationary process where the correlation length changes spatially from one set of elements to another. However, P_x is *not* guaranteed to satisfy a very important condition which is being positive definite.

Another approach to construct a non-stationary prior is to *assume independence of any two elements that belong to two different regions.* For example, in the 1D case, for two process elements x_i, x_k at locations i, k, the following covariance model

$$P(x_i, x_k) = \begin{cases} \sigma_i \sigma_k \exp\left\{-\frac{|i-k|}{l(x_i)}\right\} & \text{if } x_i \text{ and } x_k \in \Gamma(l) \\ 0 & \text{Otherwise} \end{cases} \tag{5.4}$$

will assume that x_i, x_k are independent if they belong to two different regions.

A covariance matrix P_x constructed by (5.4) is guaranteed to be a positive definite and it will represent a non-stationary process where the correlation length changes spatially from one set of elements to another. However, using (5.4) to solve large-scale estimation problems will lead to blocky artifacts in the produced estimates and associated estimation error variances. This will affect the smoothness of the resulting 2D estimates. This is because two neighbouring pixels with different correlation lengths are assumed to be independent by (5.4). Another drawback of this approach is the difficulty in defining the different region boundaries and how these boundaries change over time (expand or shrink) in the case of dynamic estimation problems.

One well-formulated approach that dealt with the non-stationarity issue which is only applicable for 1D estimation problems is based on computing some elements of the covariance and then estimating the required remaining elements based on *covariance extension and completion methods* [20, 40, 64, 79]. The covariance extension problem for a 1D process $x(i), i \in \{0, 1, \cdots, N-1\}$ is defined as follows;

Given a partial covariance matrix P_B defined as

$$P_B(t,s) = \begin{cases} p_{t,s} & \text{for some entries } (t,s) \text{ where } t,s \in \{0,1,\cdots,N-1\} \\ ? & \text{for the remaining elements} \end{cases} \tag{5.5}$$

the covariance extension problem is to find a positive-definite and symmetric covariance matrix P_C that matches the partial covariance matrix P_B at the known entries. If the goal is to have a full covariance matrix P_C that matches P_B in the known entries, then the problem is called covariance completion.

The general approach for solving the covariance extension problem is based on maximum-entropy methods [64] where a positive definite covariance matrix P_C is found such that its entropy function

$$H(P_C) = \frac{1}{N} \log |P_C| \tag{5.6}$$

is maximized. Maximum-entropy methods are *only applicable for 1D problems*. For 2D problems where the 2D covariance structure is more complicated, extending the 1D covariance extension and completion methods is a hard problem.

In dealing with this issue, we propose a method that can be considered as an alternative to the 2D covariance extension for large-scale 2D estimation problems. This method is presented in the next section.

5.4 Proposed method for non-stationary Static Estimation for 2D processes

The problem that we are trying to address is the ability to solve 2D large-scale static estimation (the update step in the dynamic estimation context) where the prior statistics are non-stationary. To clarify, consider a 2D domain with non-stationary statistics (spa-

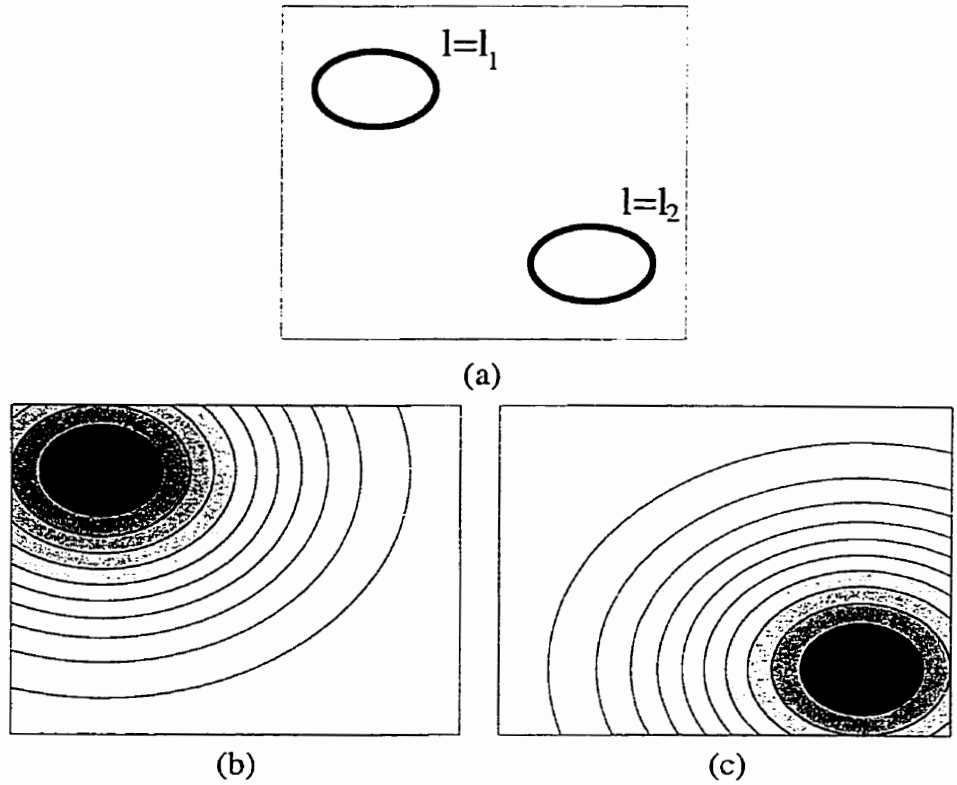


Figure 5.2: Illustration of the expected quality of the computed stationary static estimates. (a) The correlation length map of a non-stationary prior with two regions that have different correlation length $\Gamma(l_1), \Gamma(l_2)$. (b) Contours of the validity of the estimates computed based on a stationary prior with correlation length l_1 where darker colours indicate a higher quality and (c) depicts the validity of static estimates computed based on a stationary prior with correlation length l_2.

tially varying correlation length) as shown in Figure 5.2 (a) where two main regions with two different correlation lengths $\Gamma(l_1), \Gamma(l_2)$ exist. If stationary static estimates $\hat{x}(l_1)$ are computed based on a stationary prior l_1, then one should expect that the quality of the produced estimates would be high for those process elements located at $\Gamma(l_1)$ and would decay to a lower quality as one moves away from region $\Gamma(l_1)$. This is depicted in Figure 5.2(b). Similarly, Figure 5.2(c) depicts the expected validity of the estimates produced in the case that a stationary prior with correlation length l_2 is used.

The question that we try to solve is how we can compute high quality estimates in regions that have other correlation lengths l_i where $l_1 \leq l_i \leq l_2$. In other words, if two separate stationary static estimates $\hat{x}(l_1)$ and $\hat{x}(l_2)$ in addition to their associated estimation error variances $\tilde{\mathbf{p}}(l_1), \tilde{\mathbf{p}}(l_2)$ are computed how one can combine them efficiently such that the validity of the combined estimation result is *high over the whole 2D domain*. In addition, the produced estimates and their associated estimation error variances are *smoothed* everywhere, i.e., with *no blocky artifacts* generated.

Now, since our goal is to be able to perform non-stationary static estimation according to (5.1), (5.2), our approach is to express the estimates computed based on a non-stationary prior P_x as some linear combination of estimates computed based on an arbitrary number of stationary prior models with various correlation lengths l_i.

$$\hat{x}(P_x) = \alpha_x(l_1)\hat{x}(l_1) + \alpha_x(l_2)\hat{x}(l_2) + \cdots + \alpha_x(l_k)\hat{x}(l_k) \tag{5.7}$$

$$\tilde{\mathbf{p}}(P_x) = \alpha_p(l_1)\tilde{\mathbf{p}}(l_1) + \alpha_p(l_2)\tilde{\mathbf{p}}(l_2) + \cdots + \alpha_p(l_k)\tilde{\mathbf{p}}(l_k) \tag{5.8}$$

where $P(l_i)$ is an isotropic exponential prior model with correlation length l_i, and $\{l_i\}$ is a set of correlation lengths that spans the whole correlation length spectrum of the 2D process. In the following discussion we will refer to $\{l_i\}$ as interpolants. The $\alpha_x(l_i)$ are weighting coefficients associated with estimates computed based on a stationary prior l_i and $\alpha_p(l_i)$ are the weighting coefficients associated with estimation error statistics produced based on a stationary prior $P(l_i)$.

Our choice for a linear model to express the actual estimates $\hat{x}(P_x)$ and their associated estimation error variances $\tilde{\mathbf{p}}(P_x)$ is justified by the fact that the update step in the Kalman filter is linear. Therefor, using (5.7) and (5.8) to compute the approximate estimates eliminates the possibility of introducing any non-linearity to the estimator.

As just mentioned, the set of correlation lengths $\{l_i\}$ must span the whole correlation length spectrum of the 2D process. An intuitive choice for the number of interpolating stationary priors $P(l_i)$ would be just two, i.e., the smallest and the largest possible correlation lengths in the domain, l_{small}, l_{large}. However, as illustrated in Figure 5.2, this selection does not allow estimates to be interpolated properly especially for large domains where the difference between l_{small} and l_{large} is large. Therefore, incorporating a set of intermediate correlation lengths, i.e., $\{l_{small}, l_2, l_3, \cdots, l_{large}\}$, will lead to better interpolation results.

An important issue is to determine the best interpolating weights for both the state estimates α_x and for the estimation error variances α_p. This issue is discussed in the following section.

5.4.1 Determination of best interpolating weights

In this section we address the issue of determining the interpolating weights for both the state estimates α_x and for the estimation error variances α_p at each value l in the correlation range of the process.

Starting by a set of k interpolants, i.e.,

$$\{l_{small}, l_2, l_3, \cdots, l_k = l_{large}\}$$

in addition to an estimate for the predicted correlation length $L(t+1|t)$ at each pixel in the 2D domain. Our approach is to compute merged estimates at any value $l \in [l_{small}, l_{large}]$ based on the following sets of interpolants:

$\text{Set}_1 \quad : \{l_{small}, l_1, l_2\} \qquad \text{for} \quad l_{small} \leq l < l_2$

$$\begin{array}{lll} \text{Set}_2 & : \{l_2, l_3, l_4\} & \text{for } \; l_2 \leq l < l_3 \\ \vdots & \vdots & \vdots \\ \text{Set}_{k-2} & : \{l_{k-3}, l_{k-2}, l_k\} & \text{for } \; l_{k-2} \leq l \leq l_k = l_{large} \end{array}$$

Now, consider the simple case where three interpolating isotropic stationary priors are used to perform a non-stationary static estimation: $l_i < l_{i+1} < l_{i+2}$. The state estimates for those field elements that have a correlation length l such that $l_i \leq l \leq l_{i+1}$, can be written as

$$\hat{x}(l) = \alpha_x(l, l_i)\hat{x}(l_i) + \alpha_x(l, l_{i+1})\hat{x}(l_{i+1}) + \alpha_x(l, l_{i+2})\hat{x}(l_{i+2}) + e_x(l) \quad (5.9)$$

$$\tilde{\mathbf{p}}(l) = \alpha_p(l, l_i)\tilde{\mathbf{p}}(l_i) + \alpha_p(l, l_{i+1})\tilde{\mathbf{p}}(l_{i+1}) + \alpha_p(l, l_{i+2})\tilde{\mathbf{p}}(l_{i+2}) + e_p(l) \quad (5.10)$$

where $\hat{x}(l), \tilde{\mathbf{p}}(l)$ are the exact estimate and estimation error variances computed based on an isotropic stationary prior with a correlation length of l, $\hat{x}(l_i), \tilde{\mathbf{p}}(l_i)$ are the estimate and estimation error variances, respectively, computed based on isotropic prior with correlation length l_i. The $\alpha_x(l, l_i), \alpha_p(l, l_i)$ are the weights associated with estimates / estimation error variances computed based on a stationary prior with correlation length l_i used to interpolate estimates for elements with correlation length l. $e_x(l)$,$e_p(l)$ are error terms associated with expressing the exact estimates $\hat{x}(l), \tilde{\mathbf{p}}(l)$ as a linear combination of estimates computed based on other stationary priors.

In order to ensure that the interpolating weights for each interpolant $P(l_i)$ exactly add up to one and therefore exact estimates are attained at those values, $\alpha_x(l, l_{i+2}), \alpha_p(l, l_{i+2})$ are forced to be the following

$$\alpha_x(l, l_{i+2}) = 1 - \alpha_x(l, l_i) - \alpha_x(l, l_{i+1}) \quad (5.11)$$

$$\alpha_p(l, l_{i+2}) = 1 - \alpha_p(l, l_i) - \alpha_p(l, l_{i+1}) \quad (5.12)$$

Now, writing (5.9) for all the 2D field elements will give

$$\begin{bmatrix} \widehat{x}_{11}(l) \\ \widehat{x}_{21}(l) \\ \vdots \\ \widehat{x}_{n1}(l) \\ \vdots \\ \widehat{x}_{nn}(l) \end{bmatrix} = \begin{bmatrix} \widehat{x}_{11}(l_i) & \widehat{x}_{11}(l_{i+1}) & \widehat{x}_{11}(l_{i+2}) \\ \widehat{x}_{21}(l_i) & \widehat{x}_{21}(l_{i+1}) & \widehat{x}_{21}(l_{i+2}) \\ & \vdots & \\ \widehat{x}_{n1}(l_i) & \widehat{x}_{n1}(l_{i+1}) & \widehat{x}_{n1}(l_{i+2}) \\ & \vdots & \\ \widehat{x}_{nn}(l_i) & \widehat{x}_{nn}(l_{i+1}) & \widehat{x}_{nn}(l_{i+2}) \end{bmatrix} \begin{bmatrix} \alpha_x(l, l_i) \\ \alpha_x(l, l_{i+1}) \\ \alpha_x(l, l_{i+2}) \end{bmatrix} + \begin{bmatrix} e_{11}(l) \\ e_{21}(l) \\ \vdots \\ e_{n1}(l) \\ \vdots \\ e_{nn}(l) \end{bmatrix} \tag{5.13}$$

Equivalently, the above can be written in a matrix form as

$$\underline{\widehat{x}}(l) = H_x \underline{\alpha}_x(l) + \underline{e}_x \tag{5.14}$$

where $\underline{\hat{x}}$ is of size $N^2 \times 1$, H_x is of size $N^2 \times 3$, and $\underline{e}_x$ is $N^2 \times 1$. The problem now is to find the estimated weights $\{\hat{\alpha}_x(l_i)\}$ that will minimize the squared error $\| \underline{e}_x \|^2$. Observe that this is just the well-known least-squares estimation where the objective function is

$$J = \| \underline{\hat{x}}(l) - H_x \underline{\hat{\alpha}}_x(l) \|^2$$

and the solution to the above which will minimize the squared error $\| \underline{e}_x \|^2$ is given by

$$\underline{\widehat{\alpha}}_x(l) = (H_x{}^T H_x)^{-1} H_x{}^T \underline{\widehat{x}}(l) \tag{5.15}$$

Using the set of estimated weights $\{\hat{\alpha}_x(l_i)\}$, we can express the merged state estimates at correlation length l for the whole field as

$$\underline{\widehat{\widehat{x}}}(l) = H_x \underline{\alpha}_x(l) \tag{5.16}$$

Similarly, in order to obtain merged estimation error variances, the same formulation

above is applied. Rewriting (5.13) for the estimation error variances

$$\begin{bmatrix} \tilde{p}_{11}(l) \\ \tilde{p}_{21}(l) \\ \vdots \\ \tilde{p}_{n1}(l) \\ \vdots \\ \tilde{p}_{nn}(l) \end{bmatrix} = \begin{bmatrix} \tilde{p}_{11}(l_i) & \tilde{p}_{11}(l_{i+1}) & \tilde{p}_{11}(l_{i+2}) \\ \tilde{p}_{21}(l_i) & \tilde{p}_{21}(l_{i+1}) & \tilde{p}_{21}(l_{i+2}) \\ & \vdots & \\ \tilde{p}_{n1}(l_i) & \tilde{p}_{n1}(l_{i+1}) & \tilde{p}_{n1}(l_{i+2}) \\ & \vdots & \\ \tilde{p}_{nn}(l_i) & \tilde{p}_{nn}(l_{i+1}) & \tilde{p}_{nn}(l_{i+2}) \end{bmatrix} \begin{bmatrix} \alpha_p(l, l_i) \\ \alpha_p(l, l_{i+1}) \\ \alpha_p(l, l_{i+2}) \end{bmatrix} + \begin{bmatrix} e_{11}(l) \\ e_{21}(l) \\ \vdots \\ e_{n1}(l) \\ \vdots \\ e_{nn}(l) \end{bmatrix} \quad (5.17)$$

Equivalently, the above can be written as

$$\tilde{\mathbf{p}}(l) = H_p \underline{\alpha}_p(l) + \underline{e}_p \quad (5.18)$$

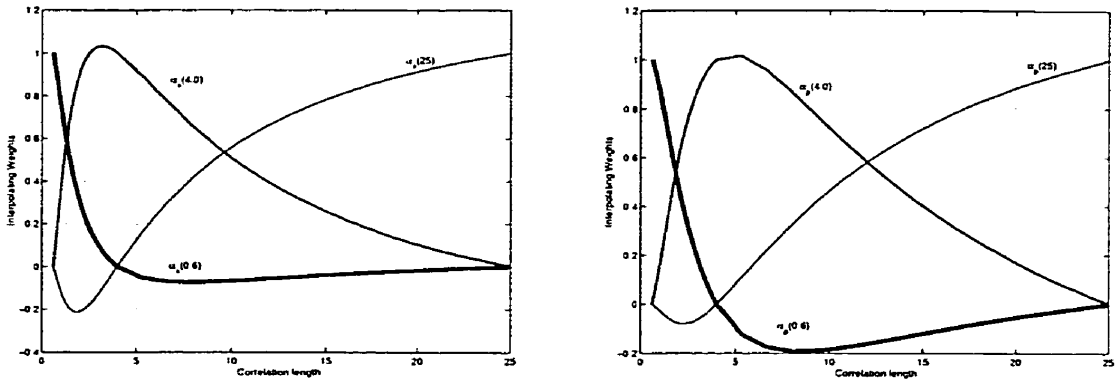
Therefore, the approximate optimal weights $\widehat{\underline{\alpha}}_p$ for the estimation error variances are given by the LLSE solution for the above overdetermined system

$$\widehat{\underline{\alpha}}_p(l) = (H_p{}^T H_p)^{-1} H_p{}^T \tilde{\mathbf{p}}(l) \quad (5.19)$$

Now, the corresponding merged estimation error variances are given by

$$\widehat{\tilde{\mathbf{p}}}(l) = H_p \underline{\alpha}_p(l) \quad (5.20)$$

Figure 5.3 depicts the general shape of the interpolating weights, for the case of a single intermediate prior. States interpolating weights α_x are shown in Figure 5.3(a) and the associated error variances interpolating weights α_p are displayed in Figure 5.3(b). In this case, the range of the correlation length spectrum is $[l_{small} = 0.6 \quad l_{large} = 25]$. An intermediate prior with correlation length $l = 4.0$ is used. In this case, we have six sets of interpolating weights: $\{\alpha_x(0.6), \alpha_x(4.0), \alpha_x(25), \alpha_p(0.6), \alpha_x(4.0), \alpha_x(25)\}$. Observe from Figure 5.3 (a,b) that by constraining the weights to add up to one according to (5.11),(5.12), their values are equal to one at their corresponding correlation lengths. In addition, each weight decays to smaller values in places that are far away from its



(a) (b)

Figure 5.3: The general shape of the interpolating weights for the states α_x (a) and the estimation error variances α_p (b). The correlation length range in this case is $[l_{small} = 0.6 \;\; l_{large} = 25]$. Interpolating priors are at correlation lengths [0.6 4.0 25]

corresponding correlation length.

An example is depicted in Figure 5.4 where five interpolating priors are used: $\{0.6, 1.5, 4, 8, 25\}$. Therefore, three sets of weights for both the states and the estimation error variances are computed:

Set1: $\{0.6, 1.5, 4\}$
Set2: $\{1.5, 4, 8\}$
Set3: $\{4, 8, 25\}$

5.4.2 Weight learning

This section presents a method for practically applying the formulation presented in the previous section. Starting with our knowledge of the 2D process correlation length range $[l_{Small} \;\; l_{Large}]$, the main issue that we try to address here is how to compute weights for the estimates and the estimation error variances at each value of $l \in [l_{Small} \;\; l_{Large}]$. In the previous section, we showed that for each l, three interpolants are used to compute the

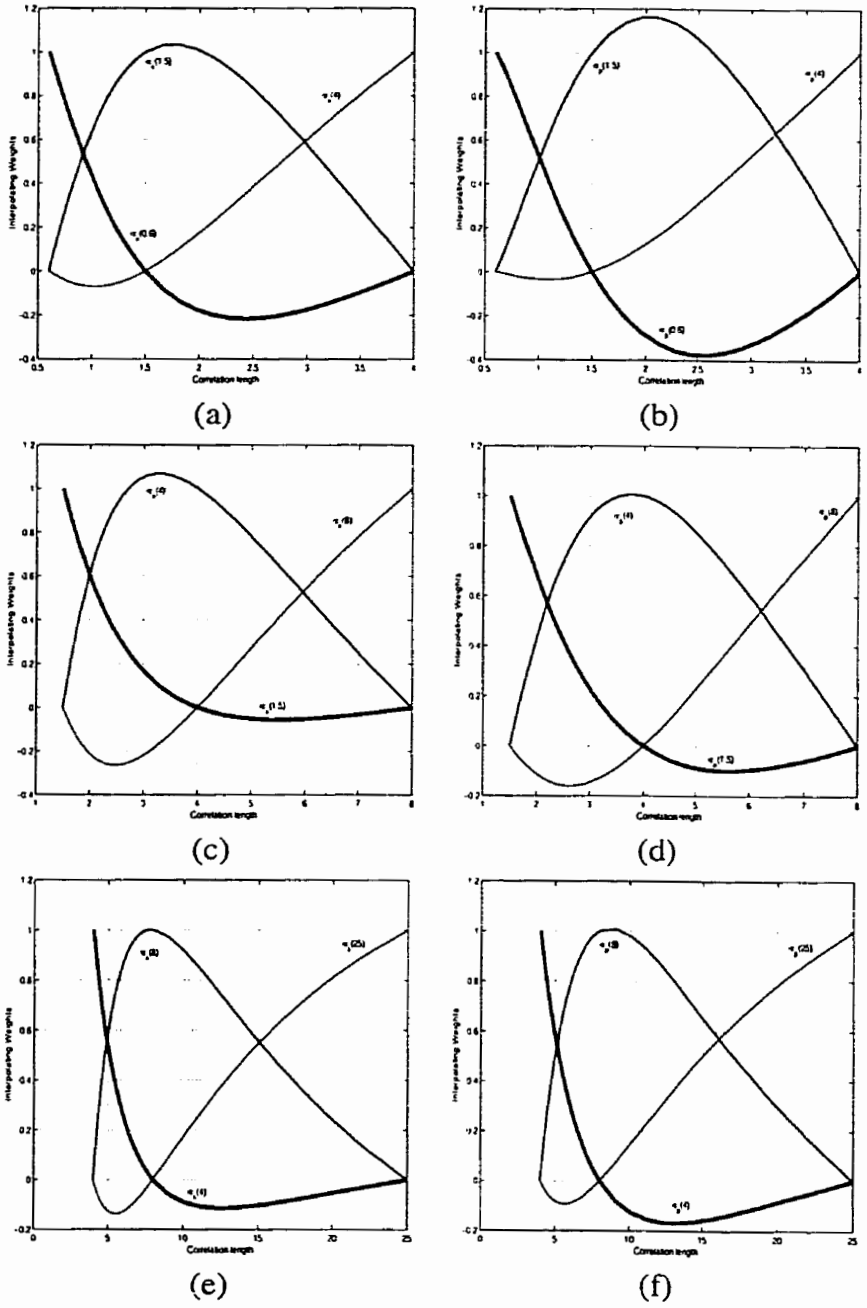


Figure 5.4: A general example showing different interpolating weights for the case of five interpolating priors at $\{0.6 \quad 1.5 \quad 4 \quad 8 \quad 25\}$. Interpolating weights for the state and the estimation error variances are shown in the left and right columns, respectively. Each figure shows the weights for a set of three consecutive and overlapped interpolating priors.

merged estimates $\hat{\underline{\hat{x}}}(l)$. The left-hand-sides of (5.14),(5.18) represent the true estimates at correlation length l and each column of matrices H_x, H_p contains true estimates computed based on each interpolants $P(l_i)$. We start by sampling the continuous correlation length range into S samples using some sampling interval. Now, the question is how to construct the left-hand-side $\underline{x}(l)$ and $\tilde{\mathbf{p}}(l)$ and matrices H_x, H_p for each sampled value l of the correlation length spectrum? Our approach is based on solving a small 2D static estimation problem exactly using (5.1), (5.2) at each sample l of the process correlation length range, i.e., $l \in [l_{Small}\ l_{Large}]$. Therefore, in order to compute the true estimates at correlation length l and the true estimates at each interpolant l_i, we should construct the following:

- A stationary prior with correlation length l representing the true process statistics and also three interpolants with correlation lengths $l_i < l_{i+1} < l_{i+2}$ for each $l_i \leq l \leq l_{i+1}$.

- Noisy measurements according to the linear measurement model

$$\underline{y} = C\underline{x}_{Truth} + \underline{v} \quad \underline{v} \text{ is} \sim \mathcal{N}(0, R) \tag{5.21}$$

Similar to Monte-Carlo simulation methods (4.10) or (4.11) are used to generate a prior $P(l)$ with correlation length l for a small size 2D process. A sample path or a realization $\underline{x}_{Truth}$ of the 2D domain $\zeta(l)$ is generated based on prior $P(l)$. This will allow us to sample noisy measurements with a measurement noise covariance R equivalent to the larger problem. In order to avoid biasing the weights, a single measurement is only sampled. Now, using the sampled noisy measurements $\underline{y}$, the measurements noise covariance R, and the prior $P(l)$, state estimates $\hat{\underline{x}}(l)$ and the associated estimation error variances $\tilde{\mathbf{p}}(l)$ are computed. Now, $\hat{\underline{x}}(l)$ and $\tilde{\mathbf{p}}(l)$ in a vector form, i.e., ordered lexicographically, will give the left hand side of (5.14) in case of the state estimates and the

left-hand-side of (5.17) in case of the estimation error variances. Similarly, each column of matrices H_x, H_p is constructed based on the position of l in the correlation length spectrum and the number of interpolating priors $P(l_i)$. If $l_i \leq l \leq l_{i+1}$ then the three consecutive priors $\{P(l_i), P(l_{i+1}), P(l_{i+2})\}$ are used to compute both the interpolating state estimates $\{\hat{\underline{x}}(l_i), \hat{\underline{x}}(l_{i+1}), \hat{\underline{x}}(l_{i+1})\}$ and the interpolating estimation error variances $\{\tilde{\mathbf{p}}(l_i), \tilde{\mathbf{p}}(l_{i+1}), \tilde{\mathbf{p}}(l_{i+1})\}$. Once $\underline{x}(l), \tilde{\mathbf{p}}(l)$ and matrices H_x, H_p are filled, then solving (5.15),(5.19) for both $\alpha_x(l)$ and $\alpha_p(l)$ is straight forward.

The complete weight learning algorithm is given in Figure 5.5.

5.4.3 Weight Optimization

The accuracy of the computed merged estimates and estimation error variances is highly dependent on two factors: the number of interpolants (K), and the correlation lengths $\{l_i\}$. The determination of optimal values for these two parameters requires setting up a performance criterion. The mean-square criterion given by

$$\text{error} = \| \hat{\underline{x}}(l) - \hat{\tilde{\underline{x}}}(l) \|^2 \tag{5.31}$$

treats all the difference between the true estimates and the approximate ones equally at each pixel of the 2D domain. This is usually not suitable in estimation problems where there exist high quality estimates having low estimation error variances and estimates with lower quality having higher uncertainties. In order to consider this fact we adopt the following criterion

$$\text{error} = \max \left(\frac{|\hat{\underline{x}}(l) - \hat{\tilde{\underline{x}}}(l)|}{\sqrt{\tilde{\mathbf{p}}(l)}} \right) \tag{5.32}$$

where $\sqrt{\tilde{\mathbf{p}}(l)}$ are the exact estimation error standard deviations. Note that the subtraction, division, and absolute value in (5.32) are pixel-wise. In Figure 5.6(a,b) the max-

1. The continuous correlation length (l), where $l \in [l_{Small}, l_{Large}]$ of the process, is descretized into S samples using some sampling interval Δl.

2. For every discrete value $l = l_{Small} + i\Delta l$, where $0 \leq i \leq S-1$, an isotropic stationary exponential prior $P(l)$ is constructed.

3. A sample path $\underline{x}_{Truth}$ is generated using the prior $P(l)$

4. Noisy measurements $\underline{y}$ are sampled randomly using

$$\underline{y} = C\underline{x}_{Truth} + \underline{v} \quad \underline{v} \text{ is} \sim \mathcal{N}(0, R) \tag{5.22}$$

5. Empirically, determine number of interpolating priors k.

6. For each value of l such that $l_i \leq l \leq l_{i+1}$ three isotropic stationary priors $\{P(l_i), P(l_{i+1}), P(l_{i+2})\}$ where $1 \leq i \leq K-2$ are constructed.

7. True estimates $\widehat{\underline{x}}(l)$ for the whole field are then computed based on the actual prior $P(l)$ for which the interpolating weights are to be learned using

$$\begin{aligned} K &= P(l)C^T(CP(l)C^T + R)^{-1} && (5.23)\\ \widehat{\underline{x}}(l) &= \underline{\mu} + K(y - C\underline{\mu}) && (5.24)\\ \tilde{P}(l) &= P(l) - KCP(l) && (5.25) \end{aligned}$$

8. Stationary states and error estimates are then computed based on the three different interpolating priors $\{P(l_i), P(l_{i+1}), P(l_{i+2})\}$ using

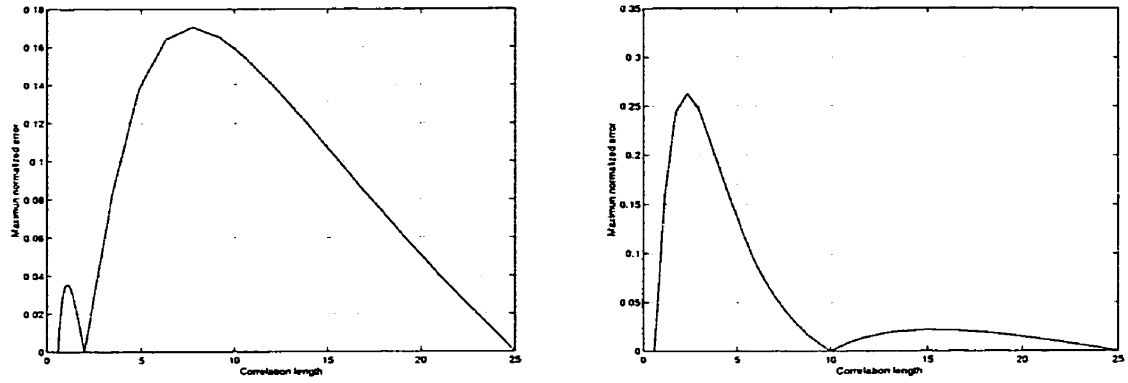
$$\begin{aligned} K &= P(l_i)C^T(CP(l_i)C^T + R)^{-1} && (5.26)\\ \widehat{\underline{x}}(l_i) &= \underline{\mu} + K(y - C\underline{\mu}) && (5.27)\\ \tilde{P}(l_i) &= P(l_i) - KCP(l_i) && (5.28) \end{aligned}$$

9. Approximate weights for the states and the error variances are then computed using (5.15),(5.19).

10. Merged estimates $\widehat{\widehat{\underline{x}}}(l)$, and merged estimation error variances $\widehat{\widehat{\mathbf{p}}}(l)$ can then be computed as

$$\begin{aligned} \widehat{\widehat{\underline{x}}}(l) &= H_x\underline{\alpha}_x(l) && (5.29)\\ \widehat{\widehat{\mathbf{p}}}(l) &= H_p\underline{\alpha}_p(l) && (5.30) \end{aligned}$$

Figure 5.5: A general procedure for generating interpolating weights for the state estimates and the estimation error variances.

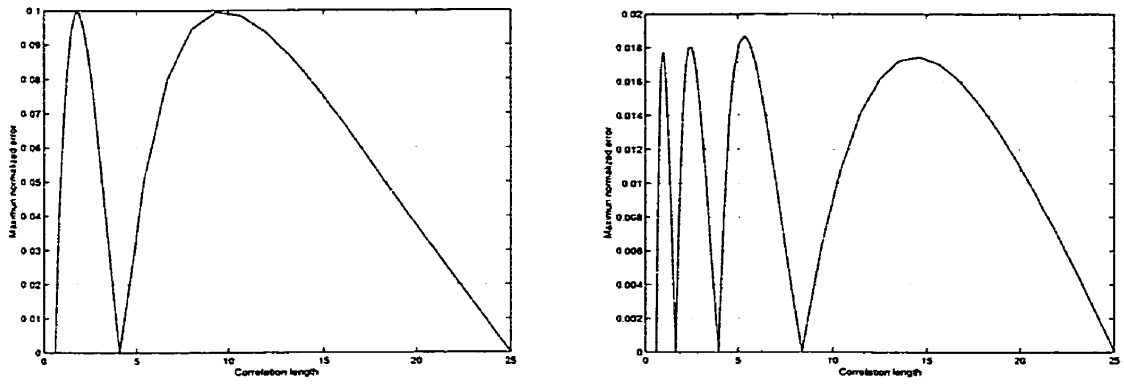


(a) (b)

Figure 5.6: Illustration of the effect of the position of the intermediate correlation length on the maximum error in the domain computed based on (5.32). In (a) the intermediate prior at $l = 2$ is positioned near $l_{Small} = 0.6$. In (b) the intermediate prior at $l = 10$ is positioned towards $l_{Large} = 25$

imum error computed according to (5.32) is plotted for all correlation length samples $l \in [l_{Small}\ l_{Large}]$. Observe that in this case we have a single intermediate prior created at two different values of l selected arbitrarily. In Figure 5.6(a), the intermediate prior is created at $l = 2$ while in Figure 5.6(b) the intermediate prior is created at $l = 10$. It is obvious that positioning the intermediate prior has a crucial effect on the maximum error.

Therefore, to get a better positioning of the *intermediate* set $l_{small} < l_i < l_{large}$ of priors correlation lengths which will minimize the maximum error over the whole correlation length range, we solve the following optimization problem



(a) (b)

Figure 5.7: Results of optimizing the values of l_i at which the intermediate priors are computed. In (a) only a single intermediate prior is positioned at $l = 4.12$ and in (b) three intermediate priors are positioned at $l_1 = 1.66, l_2 = 3.98, l_3 = 8.36$

$$\begin{aligned} \text{Minimize} &: \max\left(\frac{|\hat{\underline{x}}(l) - \hat{\tilde{\underline{x}}}(l)|}{\sqrt{\tilde{\mathbf{P}}(l)}}\right) \\ \text{Subject to} &: 0 < l_{Small} < l_i < l_{Large} \end{aligned}$$

Starting with an initial guess for the intermediate priors correlation lengths and then solving the above optimization problem using the MatLab optimization toolbox results in figuring out the best set of interpolants correlation lengths $\{l_i\}$ that will minimize the maximum error in the domain. This is depicted in Figure 5.7(a,b). Comparing Figure 5.7 (b) and Figure 5.7 (a), we observe that the maximum attained error for the optimized set of interpolants Figure 5.7(a) is (0.1) for the whole correlation length spectrum while the maximum error in Figure 5.7 (b) is (0.02) for some range of the correlation length and almost five times lower for another range.

5.5 Experimental examples

Although the main motivation of the development of the non-stationary update is to be applied in large-scale dynamic estimation problems, for the purpose of illustrating the performance we present two sets of examples: first, interpolating stationary estimates based on correlation lengths that are not used as interpolants and second, computing estimates based on non-stationary priors.

The performance will be tested on small size problems. This will allow us to solve (5.1) and (5.2) exactly and to compare the produced estimates and error variances.

5.5.1 Experiments on stationary priors

In the following examples, we apply our proposed large-scale static estimation method on 2D processes with stationary priors based on various correlation lengths that have not been used in computing the interpolating weights. The goal is to see the capability of the method to compute estimates and associated estimation error variances for various correlation lengths based on a various number of interpolating stationary priors.

The first example is based on 2D domain of size 24×24 with a correlation length of $l = 6$. Optimized interpolating weights are computed for a correlation length range of $\{l_{small} = 0.6, l_{large} = 25\}$. Estimates are computed based on a single measurement at location (12,12) with a noise variance of $R = 10$. In this experiment, three interpolants are used with correlation lengths $\{0.6, 4.12, 25\}$. Figure 5.9 depicts the obtained estimates compared to the true estimates computed by brute-force based on (5.1) and (5.2). Figure 5.9(a,c) show the actual state estimates and the associated estimation error variances, and Figure 5.9(b,d) show the interpolated or the approximate state estimates and the associated estimation error variances computed based on the interpolating weights. Observe that the obtained estimates appear to be very similar to the true estimates. By computing the normalized absolute value difference depicted in Figure 5.9(e) estimates are almost exact in the measured position and in its vicinity. However, the error in-

Number of intermediate priors	*MS error*
1	0.059
3	0.004
4	0.001

Table 5.1: Summary of the mean-square error for estimates computed based on various number of interpolating stationary priors

creases as we move away from the measured position with maximum error equals 0.094. Figure 5.9(f) depicts the absolute difference in the estimation error standard deviation. Again we see that the difference is zero at the measured position and its vicinity. Next, we repeat the same experiment but with three intermediate interpolating priors, i.e., five interpolants. The obtained estimates are depicted in Figure 5.10. The improvement in the estimates and the error variances is clear from the figure. The maximum error in the domain is 0.044 for the state estimates and 0.014 for the error standard deviation. More improvement is attained by incorporating four intermediate interpolating priors as depicted in Figure 5.11.

In Table 5.1, the mean-square error computed by

$$MSE = \frac{1}{N^2}\sum_{i,j}(\hat{\underline{x}} - \hat{\hat{\underline{x}}})^2 \tag{5.33}$$

for the estimates is listed for the above three cases. It is obvious that by incorporating more priors, the quality of the estimates and the error variances is increased at the expense of more computational demands. From the above experiments we can make the following two conclusions:

- The proposed method performs very-well (almost exactly) at the measured positions and even in the vicinity of the measured positions.

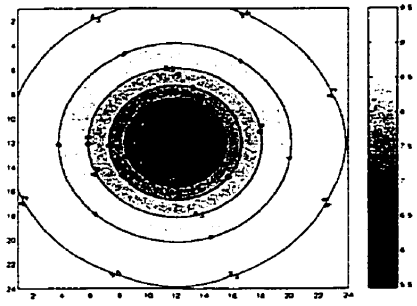


Figure 5.8: The correlation length contours of a 2D domain of size 24×24 from updating the field by a single measurement located at the center.

- The quality of the produced estimates is improved by incorporating more priors. Since for large scale problems we are employing a computationally efficient estimator, the computational requirement even by incorporating more priors is still reasonable and can easily be done on the normal workstations.

5.5.2 Experiments on non-stationary priors

In these experiments, we demonstrate the performance of our method based on a non-stationary prior. We synthesize a non-stationary covariance by using the posterior resulted from a static estimation problem,i.e., $(m_x = \hat{x}, P_x = \tilde{P})$. The original prior is stationary with correlation length equal to 10 and process variance equal to 10. The correlation structure of the posterior is depicted in Figure 5.8. Observe how the correlation length varies from small values near the center to larger values far away.

Figure 5.12 shows the estimates and the estimation error variances in the case of a *single measurement* at position $(12, 12)$. Three interpolants are used to compute the approximate estimates $\hat{\hat{x}}$. The normalized error depicted in Figure 5.12(c) is minimal at the measured position and its vicinity with a maximum value of (0.12). By increasing the number of interpolants to (6) the maximum error is reduced by a factor of two. This is depicted in Figure 5.13.

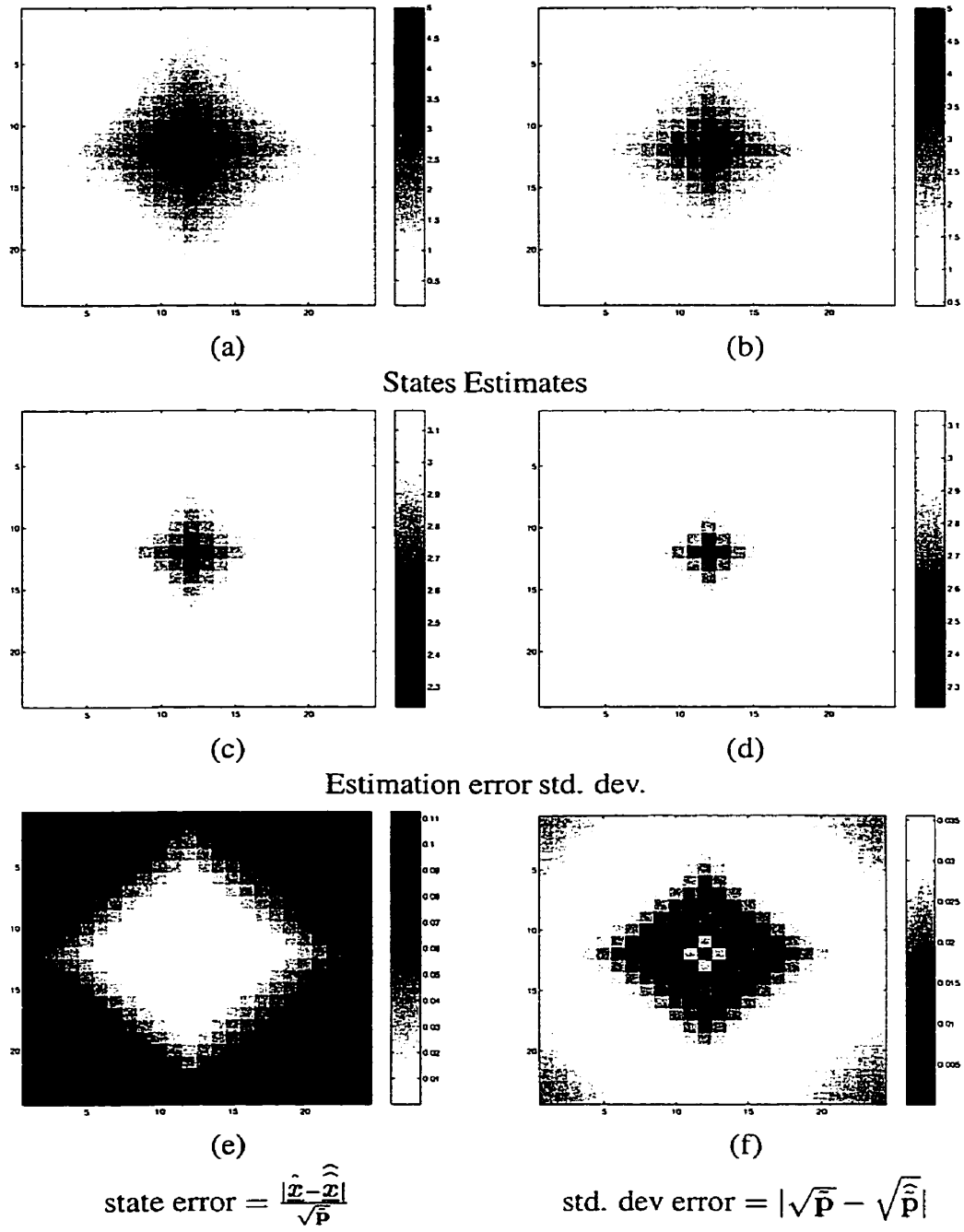


Figure 5.9: Estimation results based on a single intermediate interpolating prior for 24×24 process with correlation length equal to 6. (a) True estimates. (b) Approximate estimates. (c) True error std. dev. (d) Approximate error std. dev. (e) Normalized absolute difference. between (a) and (b). (f) Absolute difference between (c) and (d).

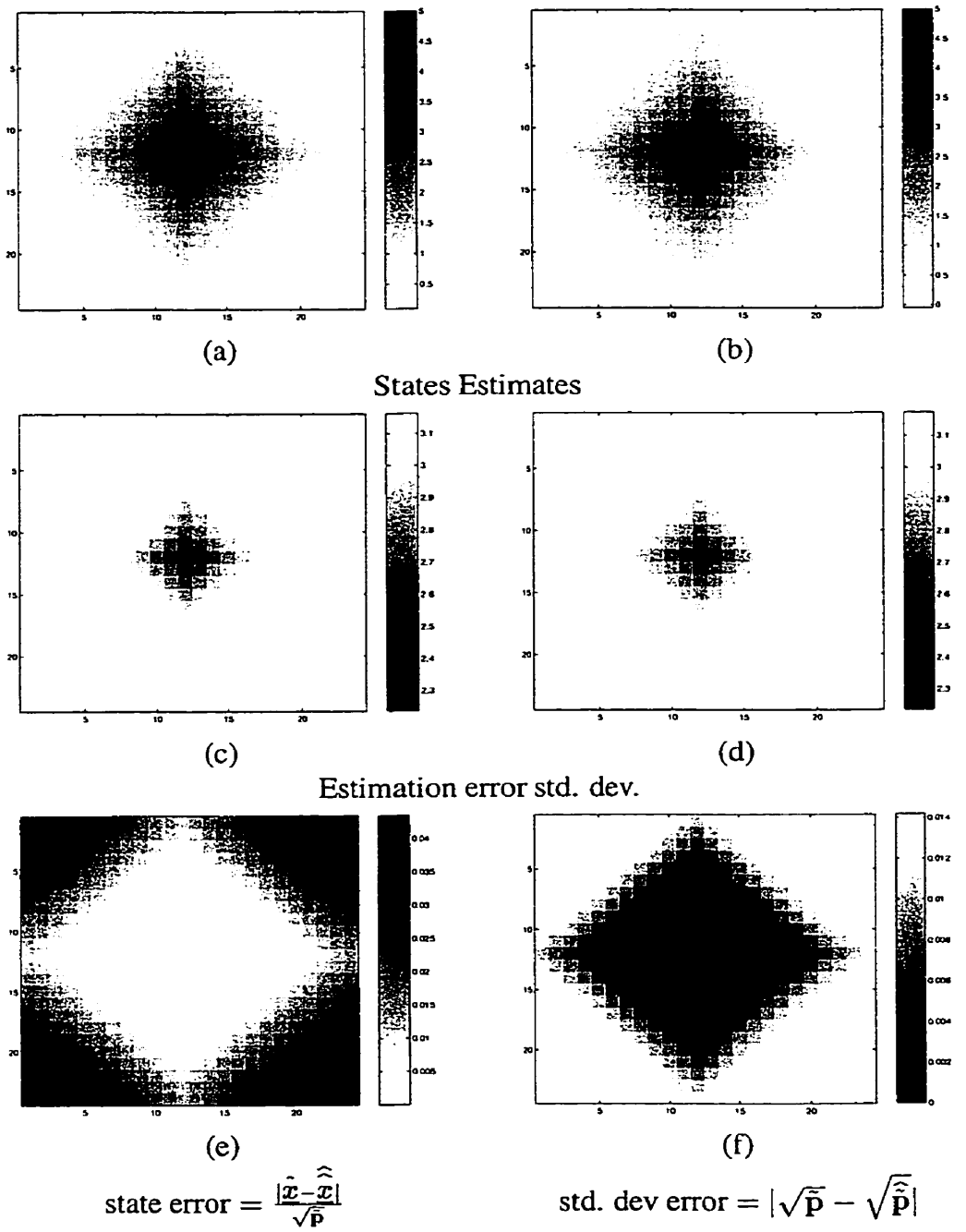


Figure 5.10: Estimation results based on three intermediate interpolating priors for 24 × 24 process with correlation length equal to 6. (a) True estimates. (b) Approximate estimates. (c) True error std. dev. (d) Approximate error std. dev. (e) Normalized absolute difference. between (a) and (b). (f) Absolute difference between (c) and (d).

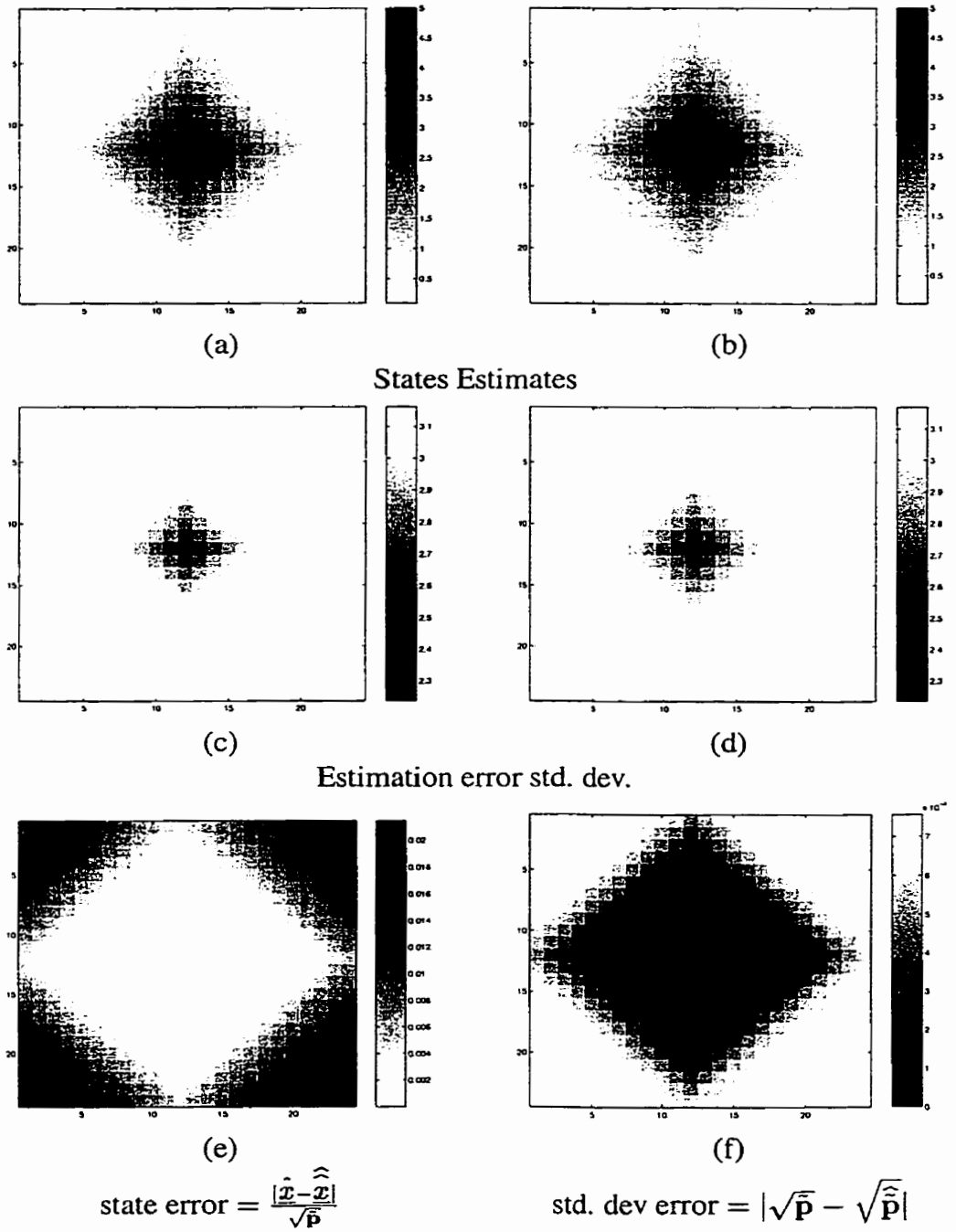


Figure 5.11: Estimation results based on four intermediate interpolating priors for 24×24 process with correlation length equal to 6. (a) True estimates. (b) Approximate estimates. (c) True error std. dev. (d) Approximate error std. dev. (e) Normalized absolute difference. between (a) and (b). (f) Absolute difference between (c) and (d).

Number of measurements	*Number of intermediate priors*	
	1	4
One	0.081	0.020
Four	0.040	0.011

Table 5.2: Summary of the mean-square error for estimates computed based on various number of interpolating stationary priors and various number of measurements

The performance is also tested for the case of *multiple measurements* at positions $[(2,2),(2,23),(23,2),(23,23)]$. Results are shown in Figure 5.14, Figure 5.15. Again we have smoothed estimates with no blocky artifacts due to the interpolation process. The estimates quality improves as we incorporate more interpolants.

A summary of the mean-square error for the estimates of the whole domain is computed for the above four cases and is given in Table 5.2.

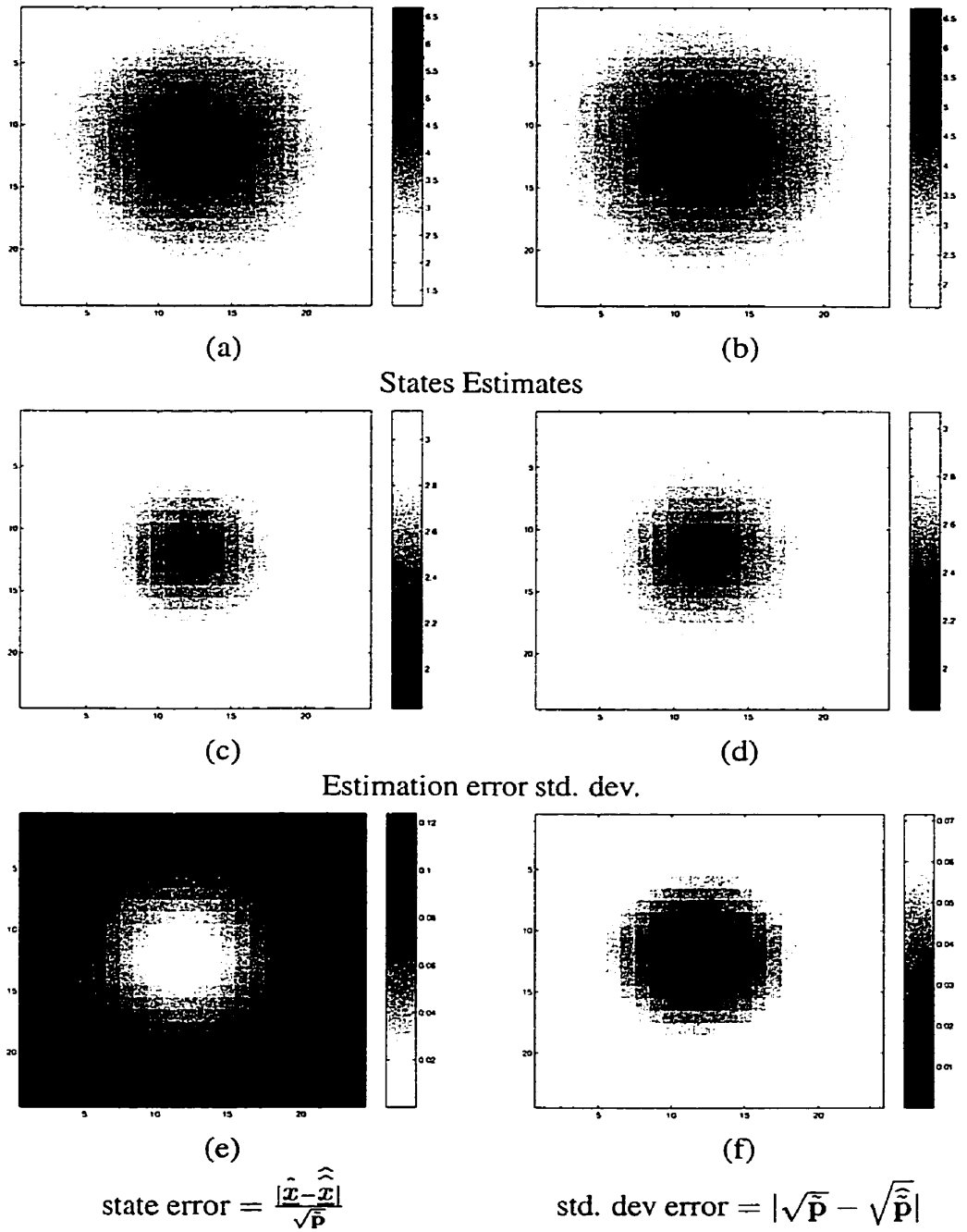


Figure 5.12: Estimation results based on three interpolants for 24 $\times$ 24 process with a single measurement at (12,12) using a non-stationary prior. (a) True estimates. (b) Approximate estimates. (c) True error std. dev. (d) Approximate error std. dev. (e) Normalized absolute difference. between (a) and (b). (f) Absolute difference between (c) and (d).

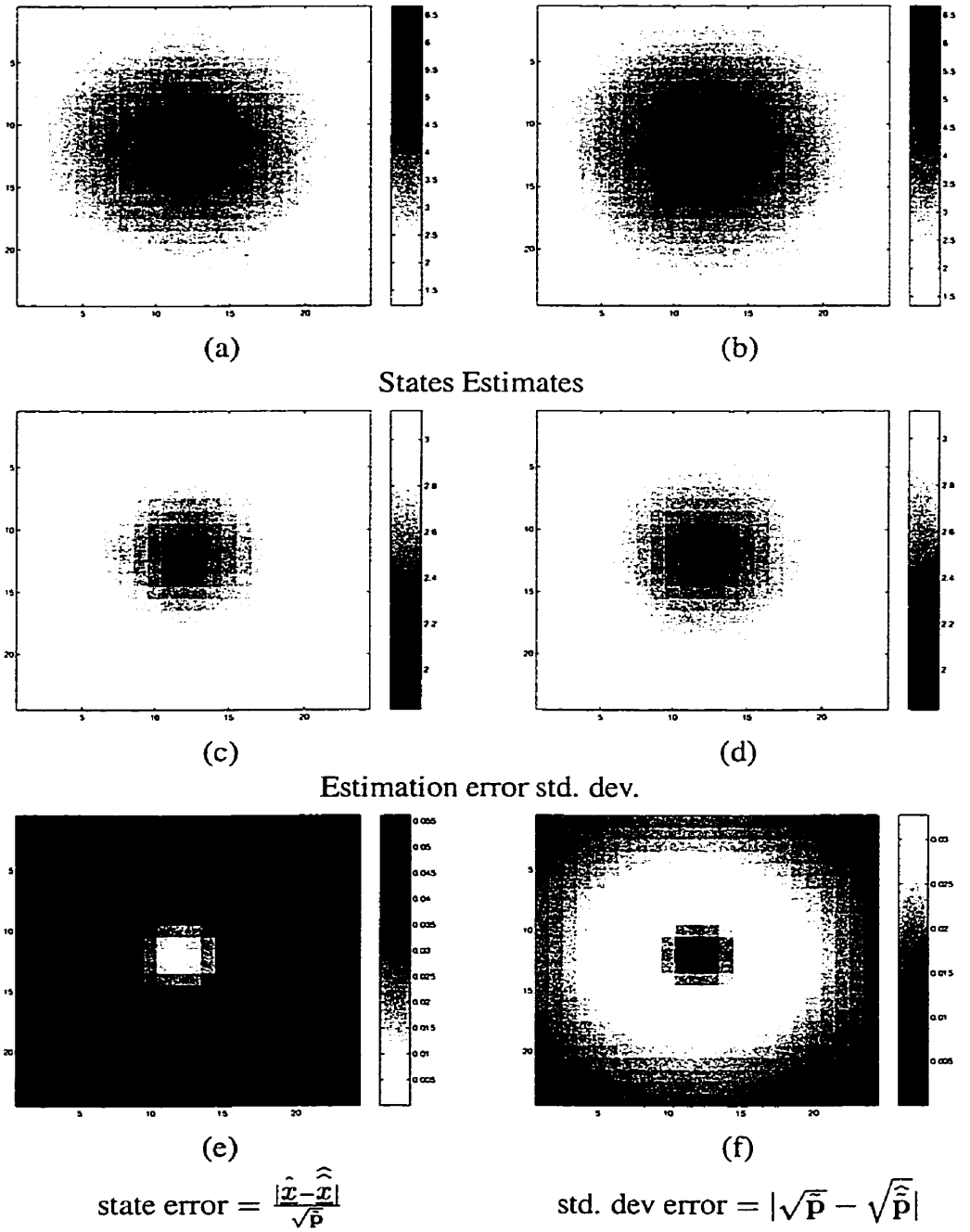


Figure 5.13: Estimation results based on six interpolants for 24×24 process with a single measurement at (12,12) using a non-stationary prior. (a) True estimates. (b) Approximate estimates. (c) True error std. dev. (d) Approximate error std. dev. (e) Normalized absolute difference. between (a) and (b). (f) Absolute difference between (c) and (d)

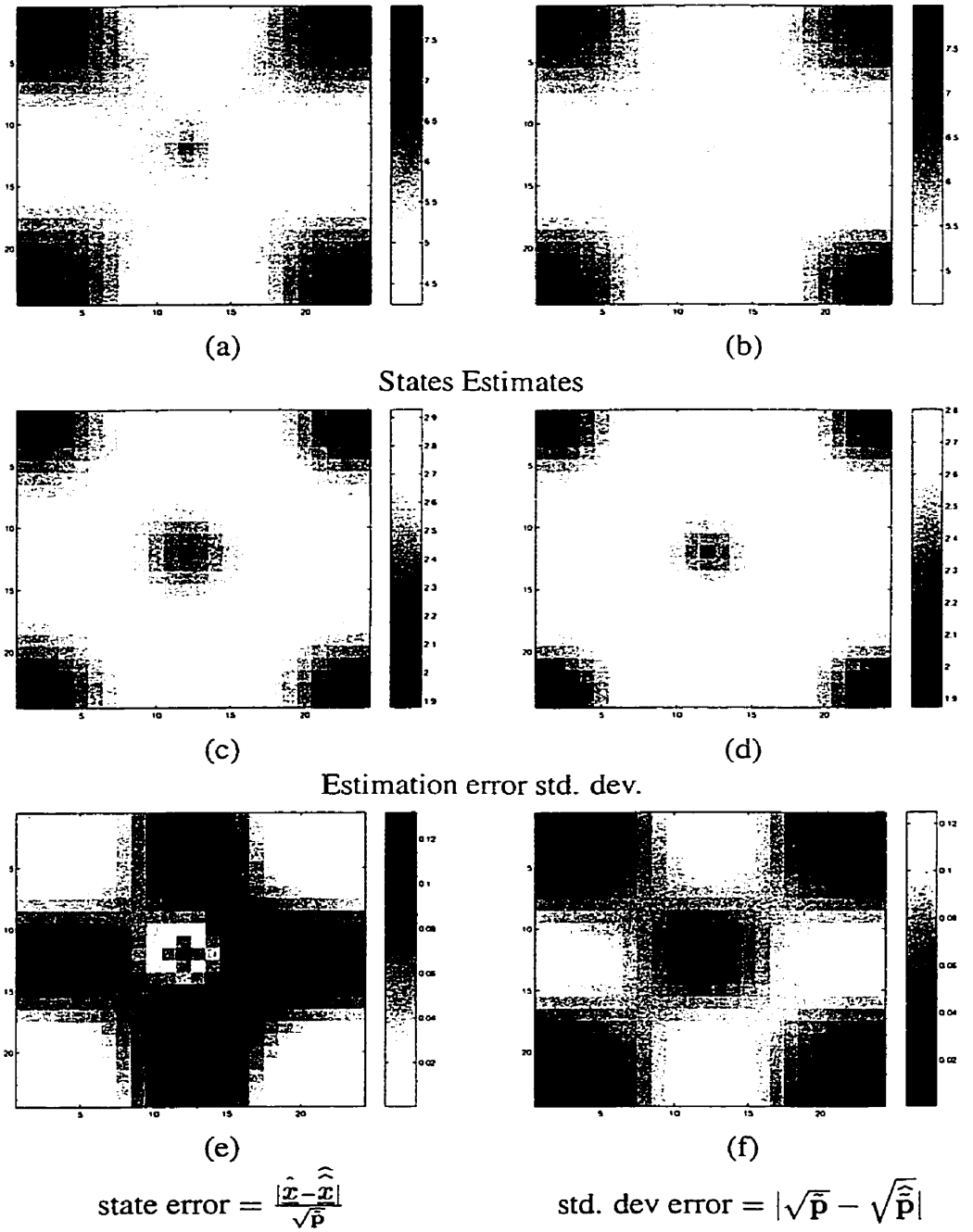


Figure 5.14: Estimation results based on three interpolants for 24 $\times$ 24 process with four measurement using a non-stationary prior. (a) True estimates. (b) Approximate estimates. (c) True error std. dev. (d) Approximate error std. dev. (e) Normalized absolute difference. between (a) and (b). (f) Absolute difference between (c) and (d)

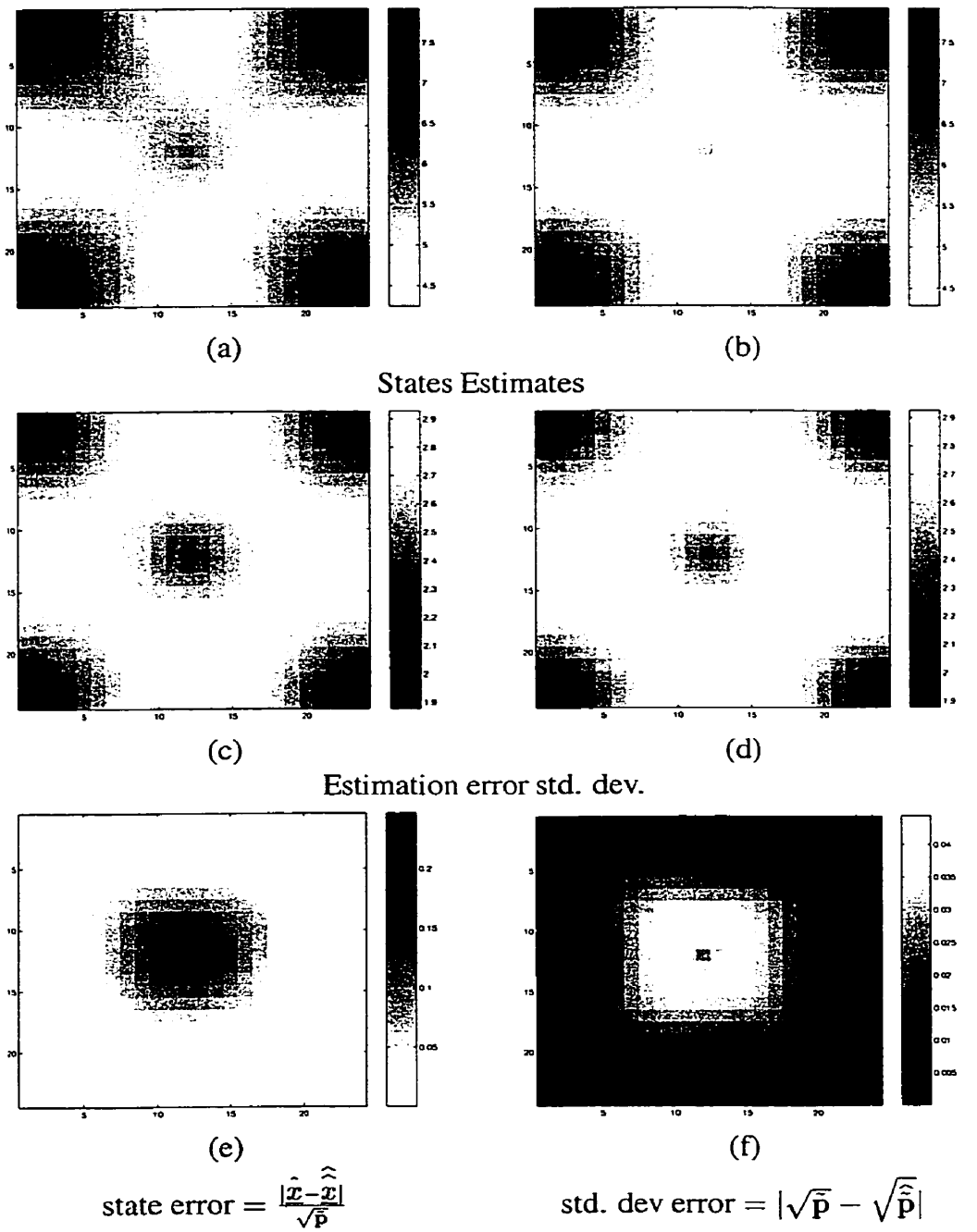


Figure 5.15: Estimation results based on six interpolants for 24 × 24 process with four measurements using a non-stationary prior. (a) True estimates. (b) Approximate estimates. (c) True error std. dev. (d) Approximate error std. dev. (e) Normalized absolute difference. between (a) and (b). (f) Absolute difference between (c) and (d)

Chapter 6

Experimental Results

In this chapter we present dynamic estimation examples in order to illustrate the performance of the proposed large-scale 2D dynamic estimator for diffusion dynamics. Two sets of examples are presented: *synthetic* problems Section 6.1 - Section 6.3 and the *ocean surface* problem Section 6.4.

For the synthetic problems, the performance of our approximate method is compared to the actual estimation results produced by the Kalman filter. For the synthetic problems, we show two types of examples: (i) time-invariant measurements and (ii) time-varying measurements. In both cases, a comparison is done between the exact solution obtained by the Kalman filter and our dynamic estimator. Both filters are initialized with the same initial conditions, i.e., $(x(0), P(0))$.

For the ocean surface problem, the multiscale estimator is used to solve the update step. In this work, no cross-validation is done because of the lack of *ground truth*, and the unavailability of exact ocean *dynamic* parameters.

6.1 Time invariant case

In the following experiments, the dynamic model is given by

$$x(t+1) = Ax(t) + w(t) \tag{6.1}$$

where A is the disceritized diffusion dynamic process. The dynamic measurement model is

$$y(t) = Cx(t) + v(t) \tag{6.2}$$

and C is kept constant over time. In other words, the number and position of the measurements are not changed overtime.

As our method is based on parameterized error models, for a given choice of diffusion A, we obtain the steady-state process covariance P_x by solving the discrete Lyapunov equation

$$P_x = AP_xA^T + Q \tag{6.3}$$

The process correlation length L is then estimated using (4.27). The filters are then initialized by

$$P(0) = \{\mathbf{p}\mathbf{p}^T\}^{\frac{1}{2}} \odot \Phi \tag{6.4}$$

where Φ is created using an isotropic stationary prior (4.10) with correlation length L and $\mathbf{p}$ are the diagonal elements (variances) of P_x. The domain size for all the presented examples is 24×24.

Interpolating weights for the state estimates (α_x) and for the estimation error variances (α_p) are computed based on the method developed in Chapter 5. These weights cover the range of the process correlation length $[l_{small} = 0.4, l_{large} = 7]$. In all experi-

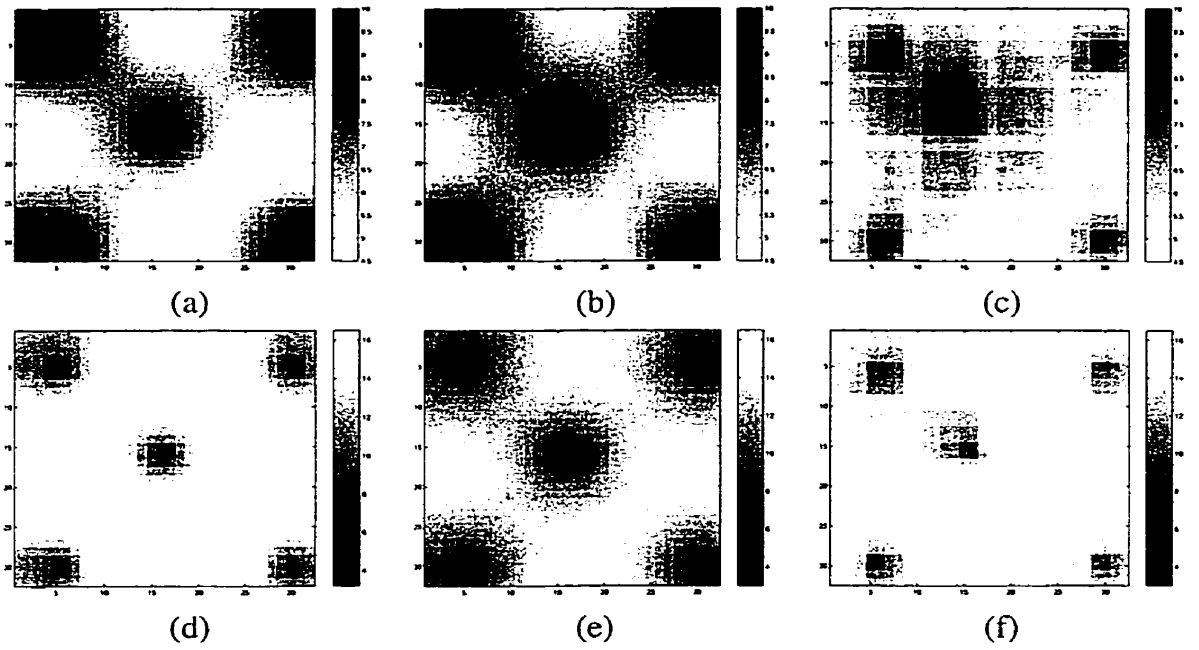


Figure 6.1: Dynamic estimates in case of five measurements after 20 update steps (a) Actual estimates. (b) Approximate estimates using exact LSE. (c) Approximate estimates using the Multiscale estimator. (d) Actual error std. (e) Approximate error std. using exact LSE. (f)Approximate error std. using the Multiscale estimator.

ments, the update step is computed based on three interpolants.

Although in all the following synthetic examples we solve the update step exactly (i.e., using brute-force matrix inversion), the estimation results in the case of small size problems are equivalent to the results obtained using the multiscale estimator. In Figure 6.1 we illustrate this fact where we compare the estimates and the estimation error standard deviations, for a 32×32 process, obtained by the Kalman filter Figure 6.1 (a,d), the approximate filter based on exact LSE Figure 6.1 (b,e), and the approximate filter using the multiscale estimator Figure 6.1 (c,f). Numerically the results obtained based on the multiscale estimator are close to the ones obtained by using LSE by brute-force except for the blocky artifacts.

6.1.1 Single measurement

The following experiment is based on a single measurement at position (12,12) sampled at each time step according to (6.2). The 2D diffusion process parameters are selected so that the steady-state process correlation length is about 7 pixels. Figure 6.2 depicts the estimation results obtained by the Kalman filter (left-hand-side) $\hat{x}(t|t)$ and by the approximate filter (right-hand-side) $\widehat{\hat{x}}(t|t)$. Panels (a),(b) show the state estimates (exact $\hat{x}(t|t)$, and approximate $\widehat{\hat{x}}(t|t)$), respectively. Observe that the approximate estimates are exact at the measured position and its neighbouring pixels. The approximate filter underestimates those elements away from the measured position. In panels (c)(exact error standard deviations $\tilde{\mathbf{p}}(t|t)$), and (d)(approximate error standard deviations $\widehat{\tilde{\mathbf{p}}}(t|t)$) we clearly see that the approximate filter underestimates the estimation error variances for elements away from the measured position due to the approximation in both update and prediction steps. In panels (e),(f) the attained error for both the estimates and the estimation error variances computed by (5.32) for the state estimates (e) and by the absolute difference for the estimation error variances (f) are shown.

6.1.2 Multiple measurements

The same setup of the previous experiment is repeated here except that each update step is based on five measurements located at (2,2) (2,23) (12,12) (23,2) and (23,23). Results are depicted in Figure 6.3. We can clearly observe that the performance of our method is quite comparable to the Kalman filter. In places where we have measurements and in their nighbourhoods we have estimates that are almost equal to the exact ones as depicted in panels (a),(b). The approximate estimation error variances, panel (d), for those places are also equal to the exact error variances, panel (c), obtained by the Kalman filter. Our method underestimates the error variances at places far from the measured positions.

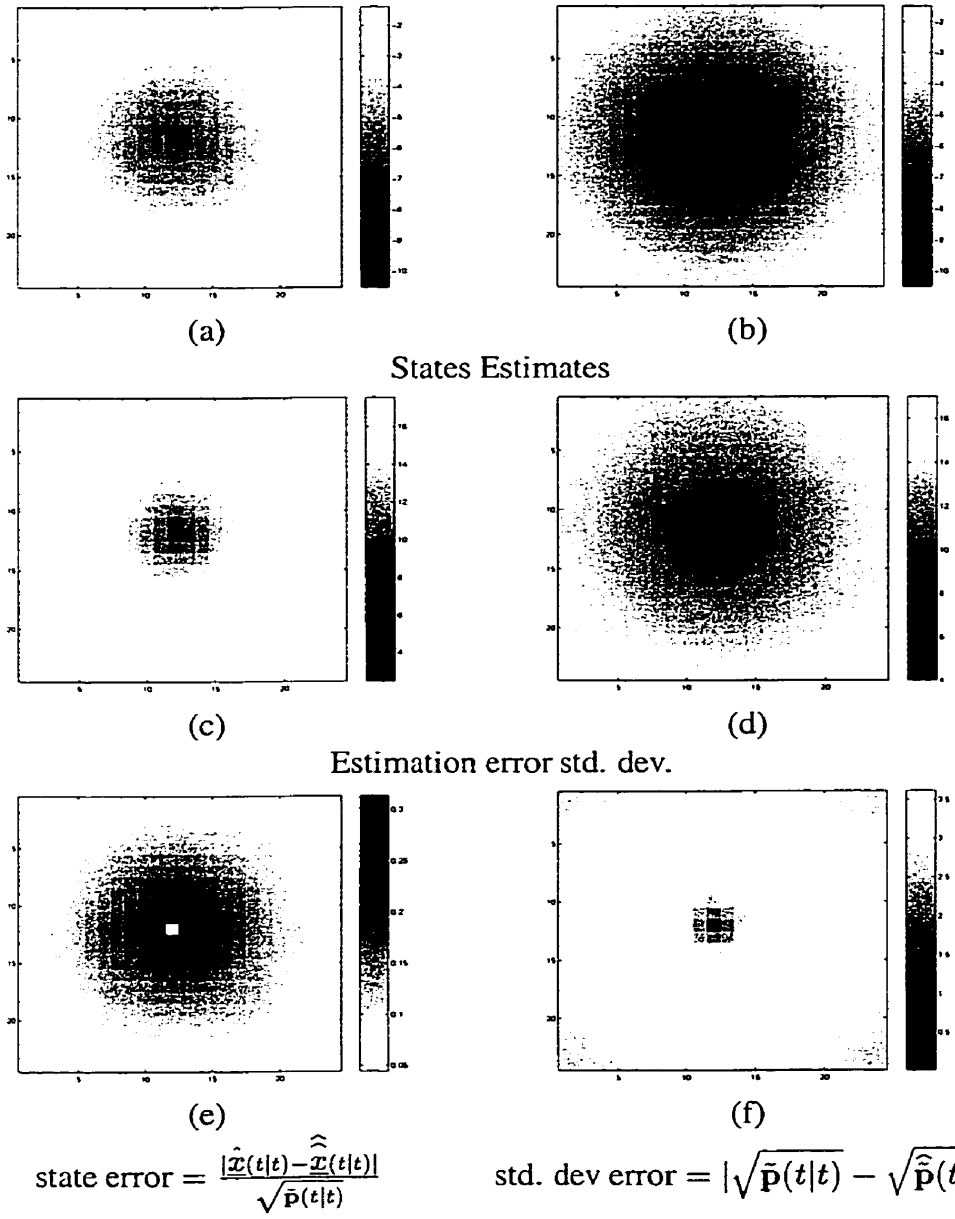


Figure 6.2: A single measurement case. Top row shows the state estimates after 20 update steps, (a) Actual and (b) Approximate. The second row shows the estimation error std. dev. (c) Actual and (d) Approximate. (e) is the normalized difference between (a) and (b). In (f) the absolute difference between (c) and (d).

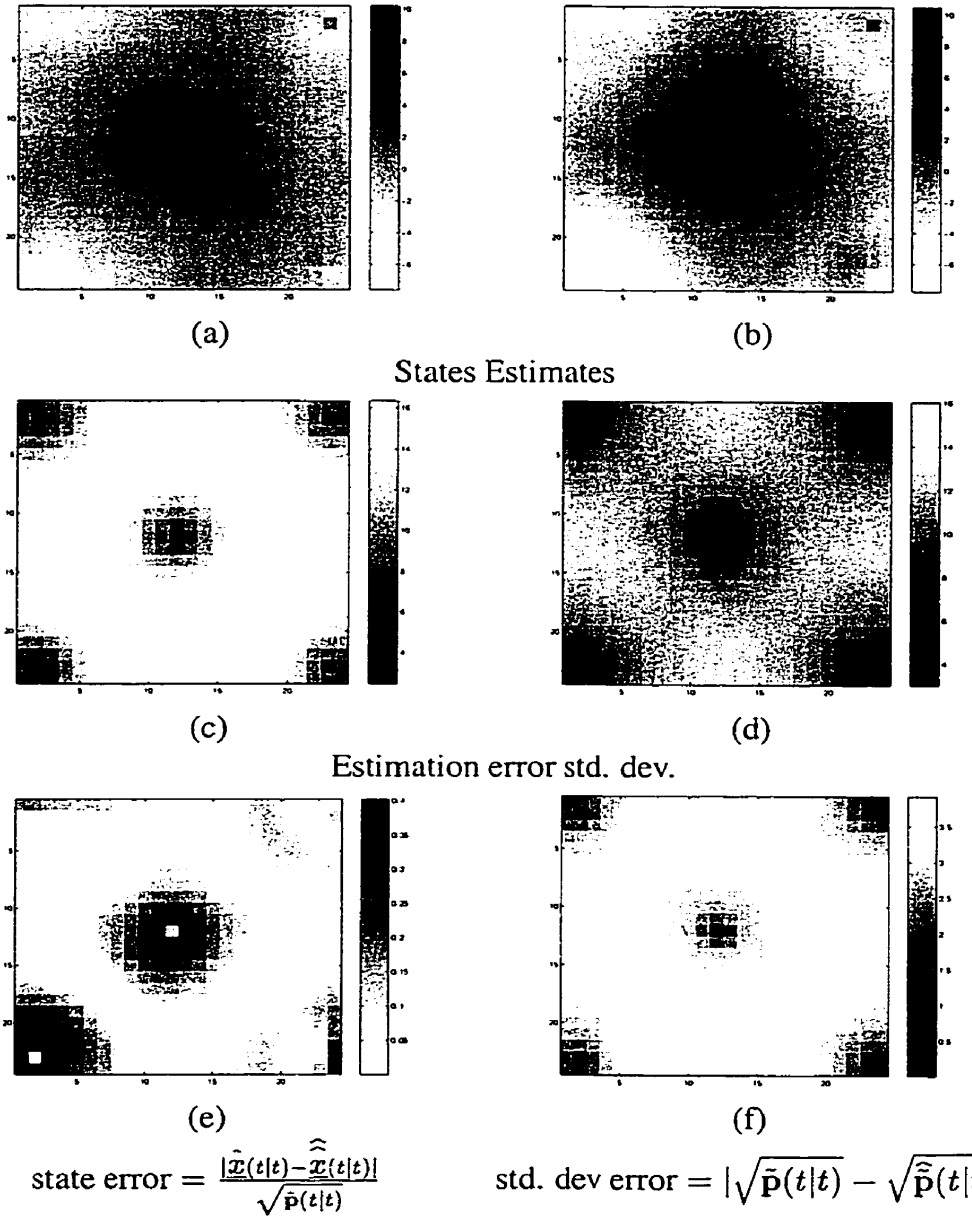


Figure 6.3: Five measurements. Top row shows the state estimates, after 20 update steps (a) Actual and (b) Approximate. The second row shows the estimation error std. dev. (c) Actual and (d) Approximate. (e) is the normalized difference between (a) and (b). In (f) the absolute difference between (c) and (d).

6.1.3 Multiple measurements with multiple predictions

The purpose of the following experiment is to see the effect of the number of predictions on the obtained estimate quality. Ten predictions between any two successive updates are computed. The performance of the approximate filter is still reasonable and it has not been greatly affected by the approximate prediction step that we adopted. The approximate estimation error variances are also comparable to the ones computed by the Kalman filter. Results are depicted in Figure 6.4.

The previous experiments are repeated but with a fixed measurement value, i.e., $y = k$ for all time steps. Although this seems unrealistic, it allows us to avoid the effect of the randomness involved in generating the measurements according to (6.1) and (6.2). The same conclusions can be reached by averaging the results of huge Monte-Carlo simulations.

In Figure 6.5 we show a 1-D profile for the estimates computed by the two filters based on a single prediction step for each update. In Figure 6.5(a) we observe that the interpolated estimates are quite good with a small difference as we move away from the measured positions. On the other hand, by increasing the number of prediction steps the difference becomes larger. This is depicted in Figure 6.5(b). In order to study the effect of prediction steps and number of measurements on the performance of the developed estimator, the following RMS of fractional error is adopted

$$\text{RMS of Fractional Error (RMSFE)} = \frac{\sqrt{\frac{1}{N^2}\sum \frac{(\hat{\underline{x}}(t|t) - \hat{\hat{\underline{x}}}(t|t))^2}{\tilde{\mathbf{p}}(t|t)}}}{\sqrt{\frac{1}{N^2}\sum \frac{\hat{\underline{x}}(t|t)^2}{\tilde{\mathbf{p}}(t|t)}}} \tag{6.5}$$

Where all operations are point-wise.

In Figure 6.6(a) the RMSFE is plotted as a function of number of updates for various number of predictions (one, five, twenty). In all experiments the update step is based on five measurements positioned at (2,2) (2,23) (12,12) (23,2) and (23,23). It is obvious that

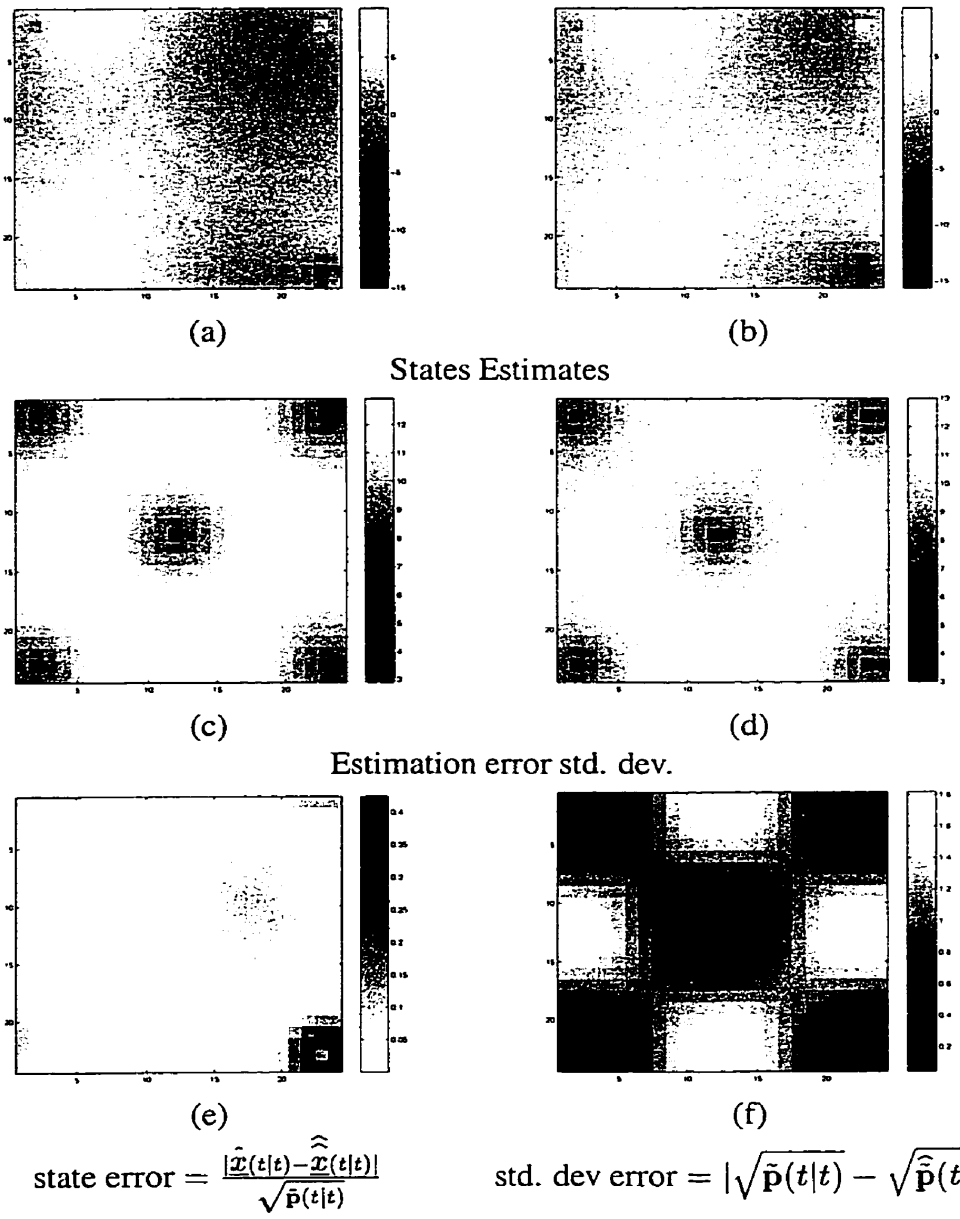
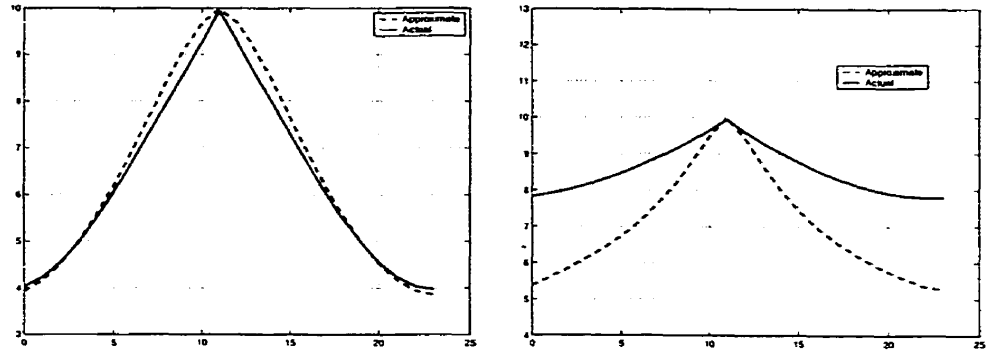


Figure 6.4: Five measurements with 10 predictions. Top row shows the state estimates after 20 update steps, (a) Actual and (b) Approximate. The second row shows the estimation error std. dev. (c) Actual and (d) Approximate. (e) is the normalized difference between (a) and (b). In (f) the absolute difference between (c) and (d).



(a) (b)

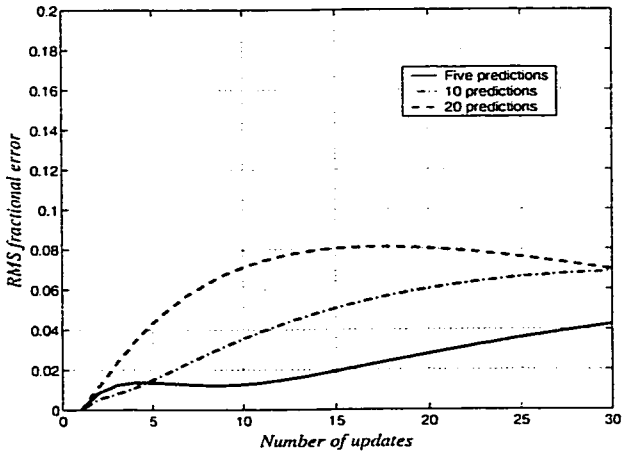
Figure 6.5: 1-D Profile for row 12 of the 24 ×24 2D diffusion process after 20 update steps with the measurement value fixed over all time steps. Actual estimates are shown as solid line and the dotted line is the approximate estimates. (a) A single prediction between each update. (b) Ten predictions are used between each update.

increasing the number of predictions magnifies the approximation effect imposed by our method.

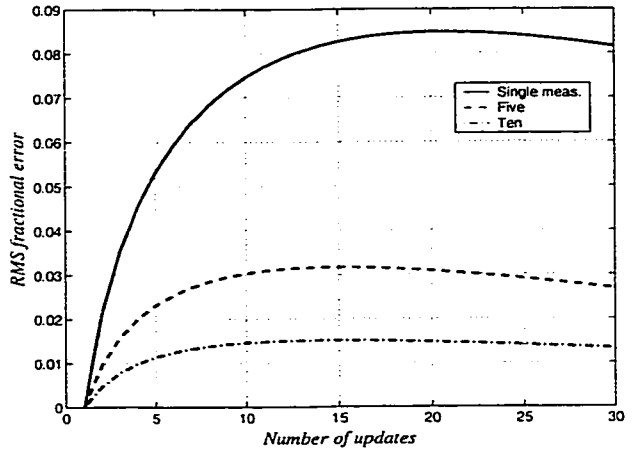
Figure 6.6(b) shows a summary for the effect of the number of measured positions on the estimates quality measured by (6.5). We conducted three different experiments based on various numbers of measurements (one, five, ten). For all experiments, a single prediction is computed for each update step. Note that if the process is weakly observed a higher RMSFE is attained.

6.1.4 Time-varying interpolating weights

The final experiment was done based on time-varying weights. In other words, more than one set of interpolating weights are used. Each set covers a different correlation length range. This is done by computing m sets of interpolating weights $\{(\alpha_{x1}, \alpha_{p_1}), (\alpha_{x2}, \alpha_{p_2}), \cdots, (\alpha_{xm}, \alpha_{p_m})\}$, each has a different value for l_{large}. At any time step, the set that has l_{large} nearest to $\max\{L(t+1|t)\}$ is used to compute the up-



(a)



(b)

Figure 6.6: Summary of the RMS fractional error computed by (6.5) for two cases: (a) The effect of number of prediction steps on the error based on observing five pixels, and (b) shows the effect of number of measurements on the error.

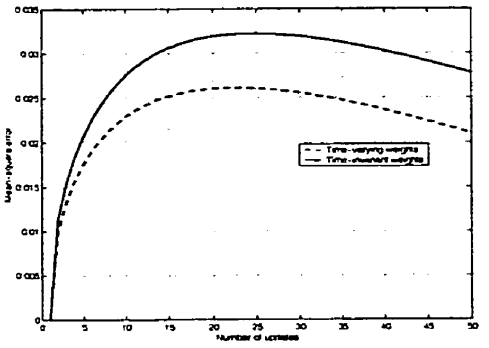


Figure 6.7: Summary of the effect of using various sets of interpolating weights on the Mean-squared error computed by (6.5).

dated estimates and their estimation error variances. Each update is based on five observations and only a single prediction between each update. Results are depicted in Figure 6.7 where we observe that by using various interpolating weights there is a slight improvement in the filter performance.

6.1.5 Convergence to steady-state

In order to empirically show that the approximate filter is capable of reaching steady-state, the filter is initialized by two different prior error variances $\mathbf{p}$ with the same correlation structure Φ. All the filter parameters are kept constant over time. In Figure 6.8, we plot the square root of the trace of the updated covariance matrix $\text{Trace}(\tilde{P}(t|t))$ at all time steps. Solid line represents the Kalman filter behaviour and the crossed line shows the approximate filter behaviour. The bold line is the solution of the Riccati equation. We observe that the approximate filter converges to steady-state, and as we expected, it converges to a lower value than the exact filter.

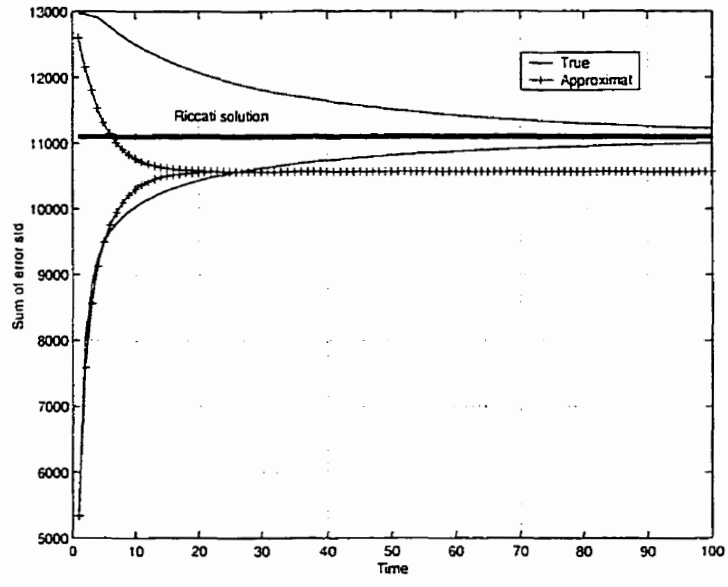


Figure 6.8: Steady state behaviour of the approximate filter. The bold line is the Riccati solution. The solid line is the Kalman filter behaviour and the crossed line is the approximate filter behaviour. The vertical axis shows the trace of the updated error covariance Trace($\tilde{P}(t|t)$)

6.2 Time-varying measurements

The experimental setup in this case is similar to the time-invariant case in Section 6.1 except that the measurements model in this experiment is given by

$$y(t) = C(t)x(t) + v(t) \tag{6.6}$$

Observe that matrix C is a function of time. This implies that at any time step, the number and positions of observations change randomly. This case is the one relevant to the large-scale problem that we are addressing which is the estimation of the ocean surface temperature based on satellite observations. In all experiments, the update step is computed based on three interpolants.

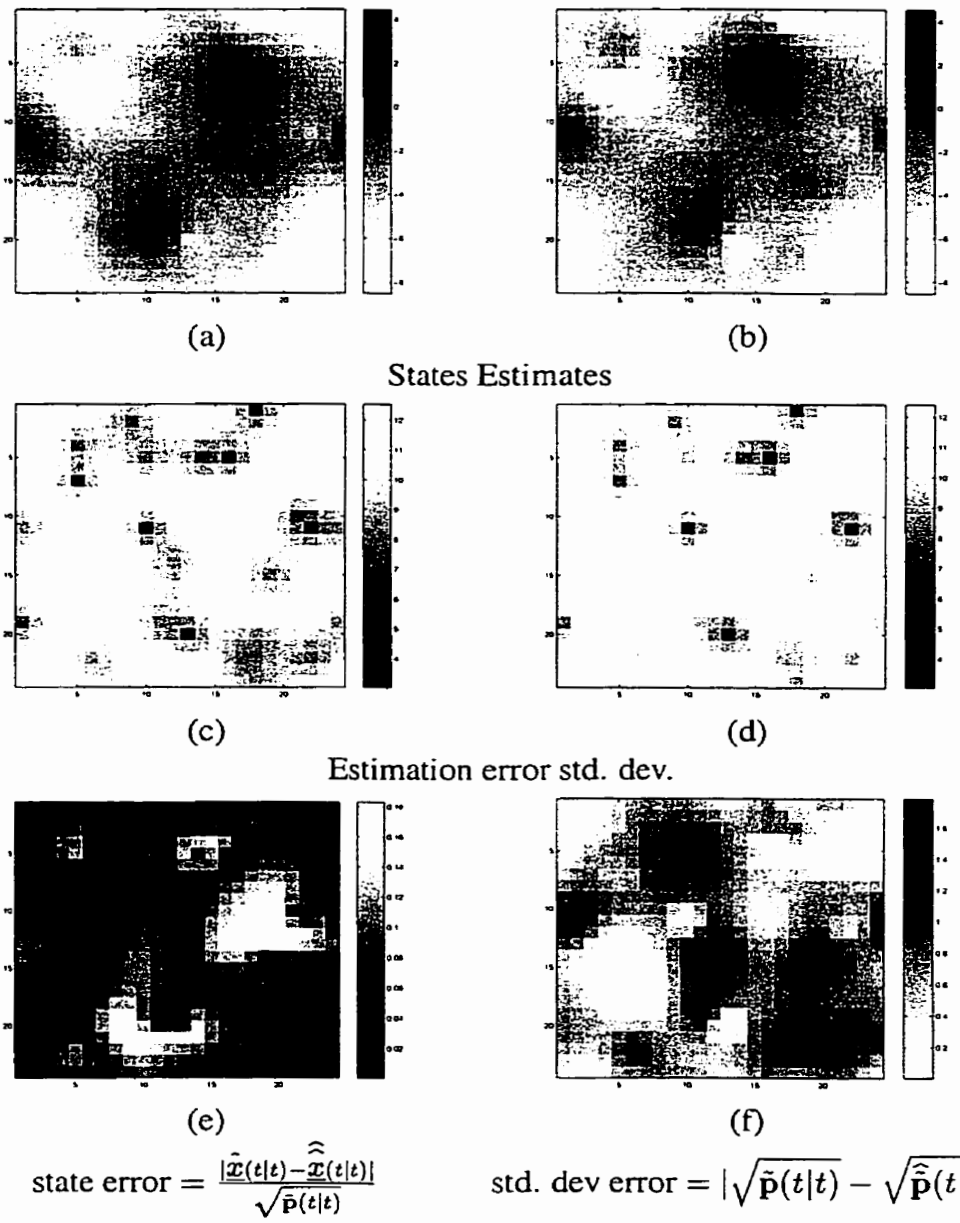
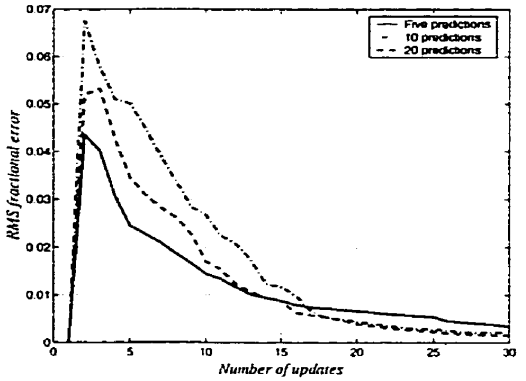


Figure 6.9: Single random measurement. Top row shows the state estimates after 20 update steps, (a) Actual and (b) Approximate. The second row shows the estimation error std. dev. (c) Actual and (d) Approximate. (e) is the normalized difference between (a) and (b). In (f) the absolute difference between (c) and (d).

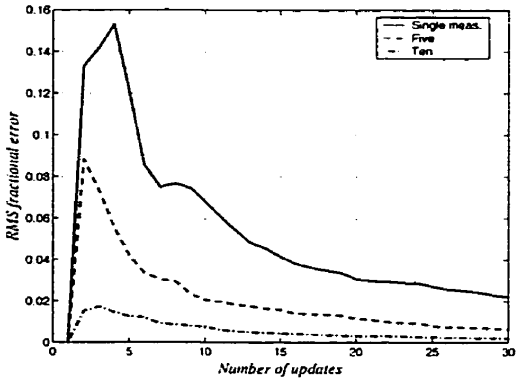
In Figure 6.9 we show an experiment where the update step is done based on a single measurement with a time-varying position. For the case of the approximate estimates, we see that the results are comparable to the actual estimates. The obtained approximate estimation error variances are also consistent with the actual. In Figure 6.9(e) we see that the maximum normalized error is (0.18) and this happens at the non-observed positions. The maximum absolute error for the estimation error standard deviations is (1.8). In order to get some notion about the effect of prediction on the estimate quality, we conducted three experiments with time-varying measurement positions but for a fixed measurement value and number. At any time step we randomly observe five pixels. The three experiments are based on a single, five, and twenty prediction steps, respectively. All these experiments are based on process correlation length of approximately seven pixels. First, in Figure 6.10 (a) we show the effect of prediction steps on the RMSFE computed by (6.5). A similar behaviour to the time-invariant case is observed. The error increases as we use more predictions steps. In this case, on the other hand, the error decays to nearly zero as number of update steps increases.

The second set of experiments are also based on process correlation length of approximately seven pixels and time-varying measurement positions but with only a single prediction step. Results are depicted in Figure 6.10 (b) where we show the effect of the number of measurements on the RMSFE. Clearly, the mean-squared error is reduced by observing more pixels.

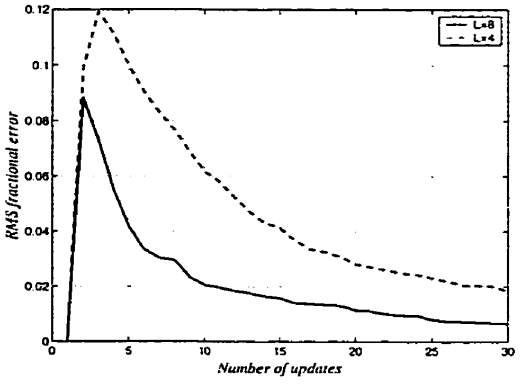
The third experiment was to study the effect of the process correlation length on the estimator performance. We conducted two experiments with time-varying measurements positions but for two different process correlation lengths $l_{large} = 4$ and $l_{large} = 8$. In these experiments we observed five pixels randomly. Figure 6.10(c) shows the resulting performance. When dealing with a process that has a larger correlation length $l_{large} = 8$, the effect of measurement is increased and the RMSFE is reduced.



(a) Effect of prediction



(b) Effect of number of measurements



(c) Effect of process correlation length

Figure 6.10: Summary of the RMS of fractional error (6.5) for time-varying measurements positions. In (a) the effect of number of prediction steps on the error. (b) the effect of number of measurements on the error. (c) the effect of the process correlation length on the error.

6.3 Comparison with the sparse Kalman filter

One common approach for solving *relatively* large-scale dynamic estimation problems is the sparse Kalman filter [4, 15, 16, 14] presented in Section 2.3.5. The computational and storage demands for this method are dependent on both the number of bands kept in the filter matrices and the number of terms used in the polynomial approximation for the matrix inversion.

The complexity of the error prediction step in the sparse Kalman filter is mainly due to the matrix multiplication operation, and it is of order $\mathcal{O}(b1 \cdot b2 \cdot n)$ where n is the field size and $b1$,$b2$ are the number of bands kept in the dynamics matrix A and the error covariance matrix $\tilde{P}(t|t)$, respectively. Our prediction approach, on the other hand, has a linear complexity (i.e., $\mathcal{O}(n)$).

The accuracy and complexity of the update step in the sparse Kalman filter depends on number of terms used in the polynomial approximation under the constraint that the matrix has to be *diagonally dominant*. By incorporating more terms in the polynomial approximation the matrix inverse converges to the true inverse with additional computational and storage demands. In our approach, the computational complexity of a single stationary update step based on the multiscale estimator is $\mathcal{O}(n^{\frac{5}{2}})$. The total computational complexity of the update step depends on number of interpolants used to compute the approximate estimates.

As an example, we compared the estimation results for the case of a 24×24 process obtained by the sparse Kalman filter and our approximate method. For this example a single pixel at (12,12) is observed. Error covariance matrices are asserted to have 29 bands which represent the nescessary bands for six-order neighbourhood structure. The update step in our method is based on three interpolants.

Results are depicted in Figure 6.11 after 50 update steps. Exact state estimates and estimation error standard deviations by Kalman filter are depicted in Figure 6.11 (a,b), respectively. Estimates computed based on parameterized error covariances are depicted

in Figure 6.11 (c,d). Figure 6.11 (e,f) show the estimates obtained by the sparse Kalman filter. For this snapshot, our method clearly outperforms the sparse Kalman filter estimates and estimation error variances. Observe the effect of imposing a local neighbourhood structure in case of the sparse Kalman filter on the estimates quality. While our approximate method produces smooth estimates that cover the whole domain, the sparse Kalman filter estimates are just local to the extend defined by the imposed neighbourhood structure.

6.4 Ocean Surface Problem

The main motivation of this research is to apply it in solving large-scale 2D dynamic estimation problems.

Recently, there has been great interest in studying the ocean surface temperature due to its relationship to climate changes. As reported by [76], the following facts motivate scientists to study the sea surface temperature (OST):

- OST plays an important role in determining the heat flux between the oceans and the atmosphere and is a major component of the global climate.
- OST is one of the most important geophysical parameters in climate studies, as the behaviour of the atmosphere is strongly coupled to the ocean temperature.
- Monitoring OST facilitates in the early detection of short-term climate anomalies such as El Nino. In addition, a current issue which is the prediction of global warming is investigated through monitoring OST.

Observations of OST are provided by a series of instruments called the Along-Track Scanning Radiometer (ATSR), infrared instruments, mounted on the ERS-1/2 and ENVISAT research satellites [75, 76]. The observed OST is highly accurate with an accuracy of 0.3 K due to the unique scanning geometry of ATSR. ATSRs view the same

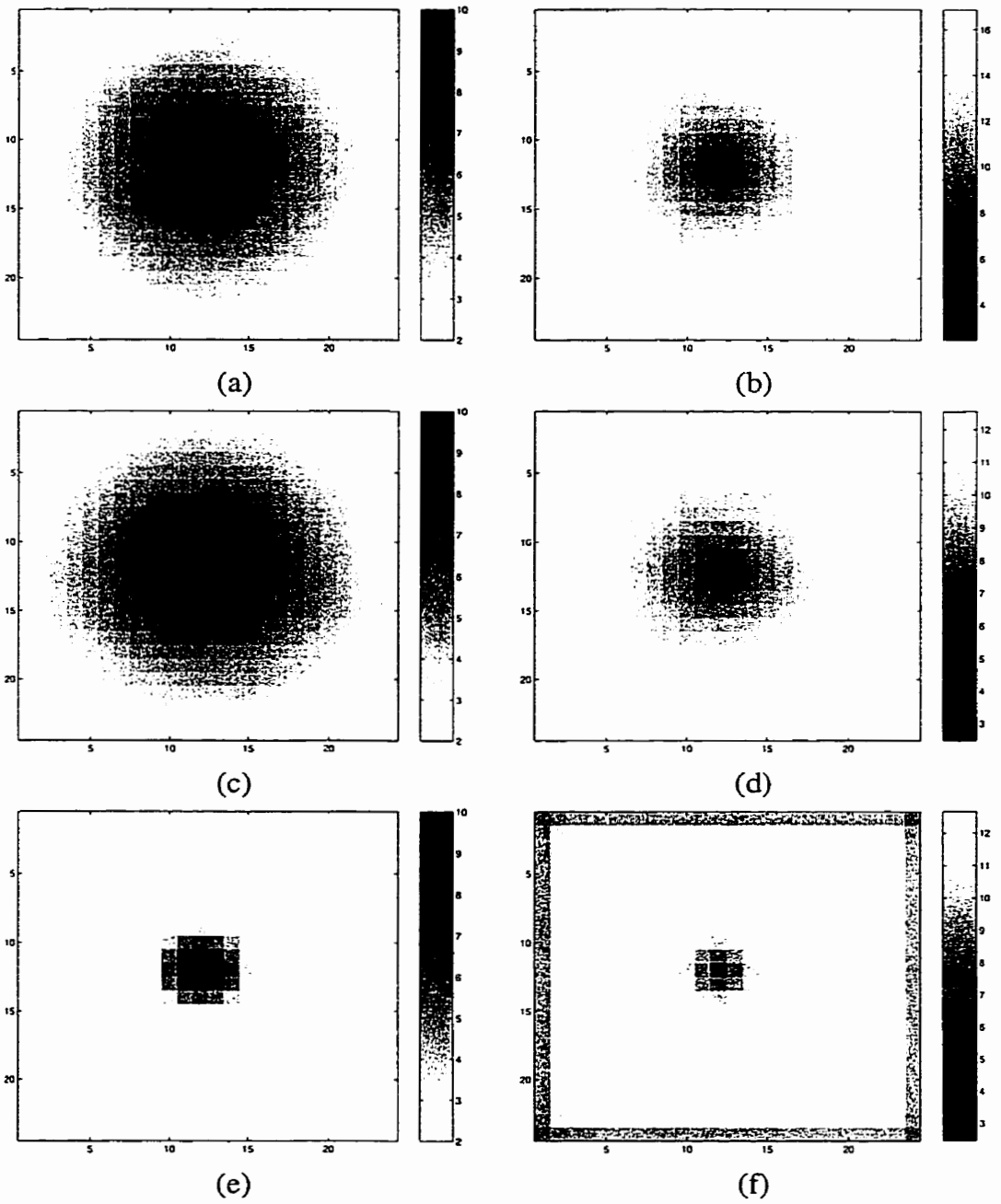
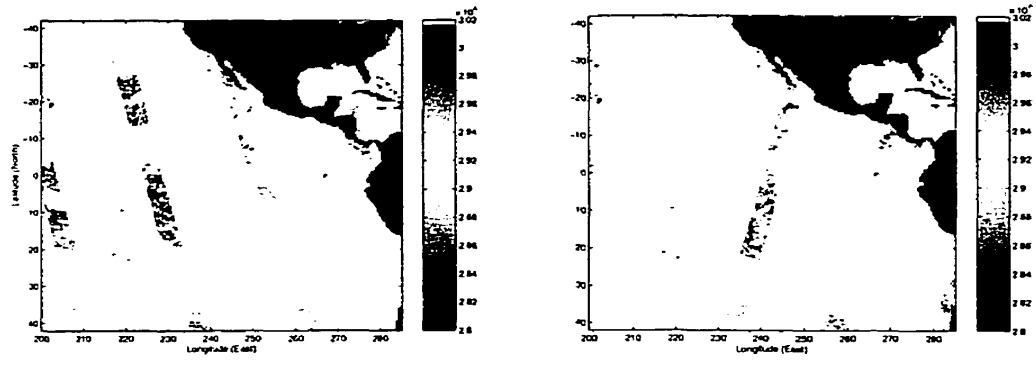


Figure 6.11: Comparison with sparse Kalman filter in case of a single time-invariant measurement. (a) exact estimates and (b) estimation error standard deviations obtained by the Kalman filter after 50 update steps. (c) and (d) depict the approximate estimates and the approximate estimation error std. based on parameterized error models. (d) and (f) are the results computed by the sparse Kalman filter with six-neighbourhood structure.



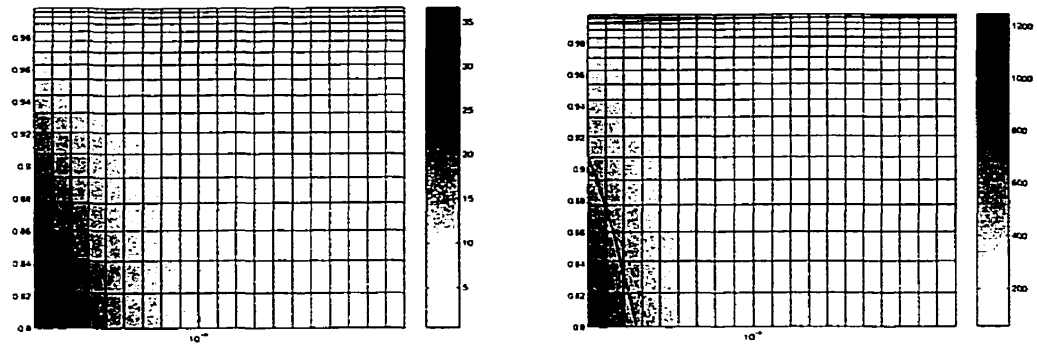
(a) (b)

Figure 6.12: Samples of the OST observations for the month of October. (a) Night data. (b) Day data

point on the sea surface twice along two different atmospheric paths allowing for the correction of any atmospheric interferences.

Figure 6.12 shows samples of the data that we are dealing with. OST observations are provided over two time periods: night time Figure 6.12 (a) and day timeFigure 6.12 (b). The size of each image is 512×512. Observe that the data is very sparse. There are many instances where no data is available. Also, there are many cases where data is missing due to clouds or due to the scanning devices.

Interpolated static estimates based on three successive observations of the ocean surface temperature have been studied before [34] based on the assumption that the ocean surface is static over a three-day period. However, the sparse nature of the data in addition to the dynamic nature of the ocean surface make solving such a problem in the dynamic estimation context a more realistic approach, yet challenging, due to the problem size. As previously elaborated, dynamic estimation allows extracting as much information as possible from all available data. In other words, estimates at any time are not only based on the available observation at the current time but also on all available observations up to the present time. In addition, in the case of no observations, available estimates are



(a) (b)

Figure 6.13: Effect of the diffusion parameters on the desired process spatial and temporal correlation length. The horizontal axis represents the heat loss parameter $b.\Delta t$ and the vertical axis represents (α) in (4.15) (a) Process spatial correlation length in pixels(b) Process temporal correlation length in time steps.

still computed based on the dynamic model. This implies that highly smoothed estimates are obtained with a lower uncertainty as more observations are incorporated.

6.4.1 Ocean surface dynamics

In this work, we model the dynamics of the ocean surface temperature as a 2D diffusion process. We want to emphasize that that the ocean dynamics are more complex than a simple diffusion model. The availability of the ocean surface temperature data is not enough to infer an approximate model for the ocean surface dynamics for at least the following reasons:

- Ocean dynamics depend on several natural factors that need to be observed in order to infer a reasonable dynamic model.
- The available observations are for the surface temperature and it is not clear how to determine other variables that have not been observed.
- The temperature measurements are local to the surrounding region and can not be

generalized to all ocean regions.

Exact modeling of the ocean surface dynamics is beyond the scope of our work. The reader is referred to [11, 12, 73, 97] for more material on ocean dynamics.

Sense the scope of this work is related to the diffusion dynamics, we model the dynamics of the ocean surface temperature as a 2D diffusion process. Although our model is not exact, it is a reasonable assumption that allows us to illustrate the capability of the developed method.

Based on our choice of modeling the ocean as diffusive according to

$$\frac{\partial T(x,y,t)}{\partial t} = a_x \cdot \frac{\partial^2 T(x,y,t)}{\partial x^2} + a_y \cdot \frac{\partial^2 T(x,y,t)}{\partial y^2} - b \cdot T(x,y,t) + \gamma \cdot w(x,y,t) \quad (6.7)$$

Where $T(x,y,t)$ is the temperature at position (x,y) and time t, and $w(x,y,t)$ is a Gaussian white noise with variance γ^2. We empirically tried to fit a reasonable discretized diffusion parameters, i.e., β, α in (4.15), by considering the ocean temporal and spatial correlation structure. A given spatial correlation length can be accomplished by various combinations of the discretized diffusion process parameters. However, by incorporating the ocean temporal correlation length a unique set of parameters can be obtained. This is illustrated in Figure 6.13 where in (a) the contour of all the diffusion parameters that can lead to a spatial correlation length of 20 pixels is plotted (the solid line in the left corner of the figure). In Figure 6.13 (b) we superimposed the contour of all diffusion parameters that lead to a time step of approximately 500 predictions (the dotted line) over the previous contour. The point of intersection gives the unique diffusion parameters that matches our choice of temporal and spatial correlation lengths.

To get an approximate notion for the ocean temporal correlation length, we used the available observations to empirically compute the sample correlation coefficient according to

$$R(l) = \frac{1}{N-l} \sum_{k=0}^{N-l-1} T_{ij}(k) T_{ij}(k+l) \quad (6.8)$$

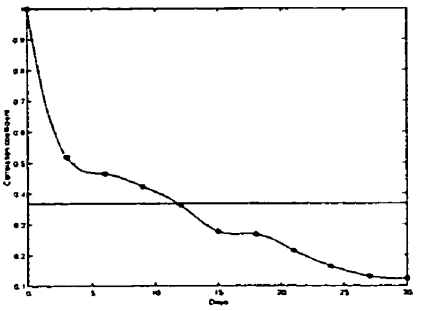


Figure 6.14: Empirical ocean temporal correlation.

Where T_{ij} is the ocean temperature at the (i, j) pixel and N is the total number of temperature frames and l is temporal separation. Figure 6.14 depicts the ocean temporal correlation.

6.4.2 OST simulation results

The developed method has been applied to a five-month period of OST observations. The diffusion parameters are chosen such that the process spatial correlation length is 20 pixels (5 degrees). The corresponding temporal correlation length is about 30 prediction steps. The initial prior is an isotropic exponential with correlation length 20 pixels and standard deviation of 70 K. The obtained temperature estimates are given in Figure 6.15 and their corresponding estimation error standard deviations are depicted in Figure 6.16. By visually inspecting the obtained estimates we can observe the following:

- The estimated images are dense and smooth with no blocky artifacts arising from the estimation process. This is because our dynamic estimation method does not compute estimates based on statistically uncorrelated regions.
- The temperature variations are compatible, to first-order, with oceanographic expectations, although further work is required here.

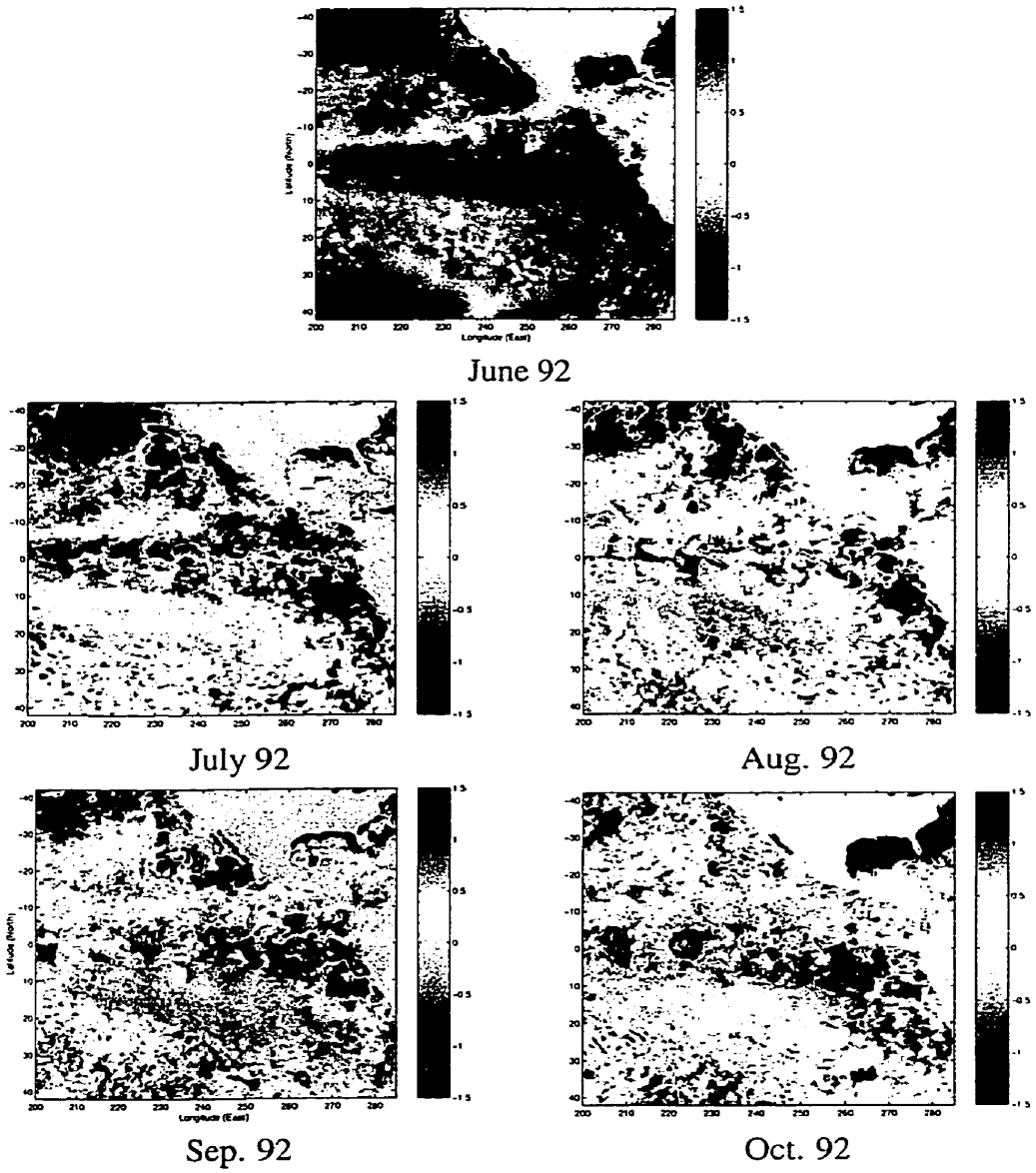


Figure 6.15: Anomaly state estimates over a five months period

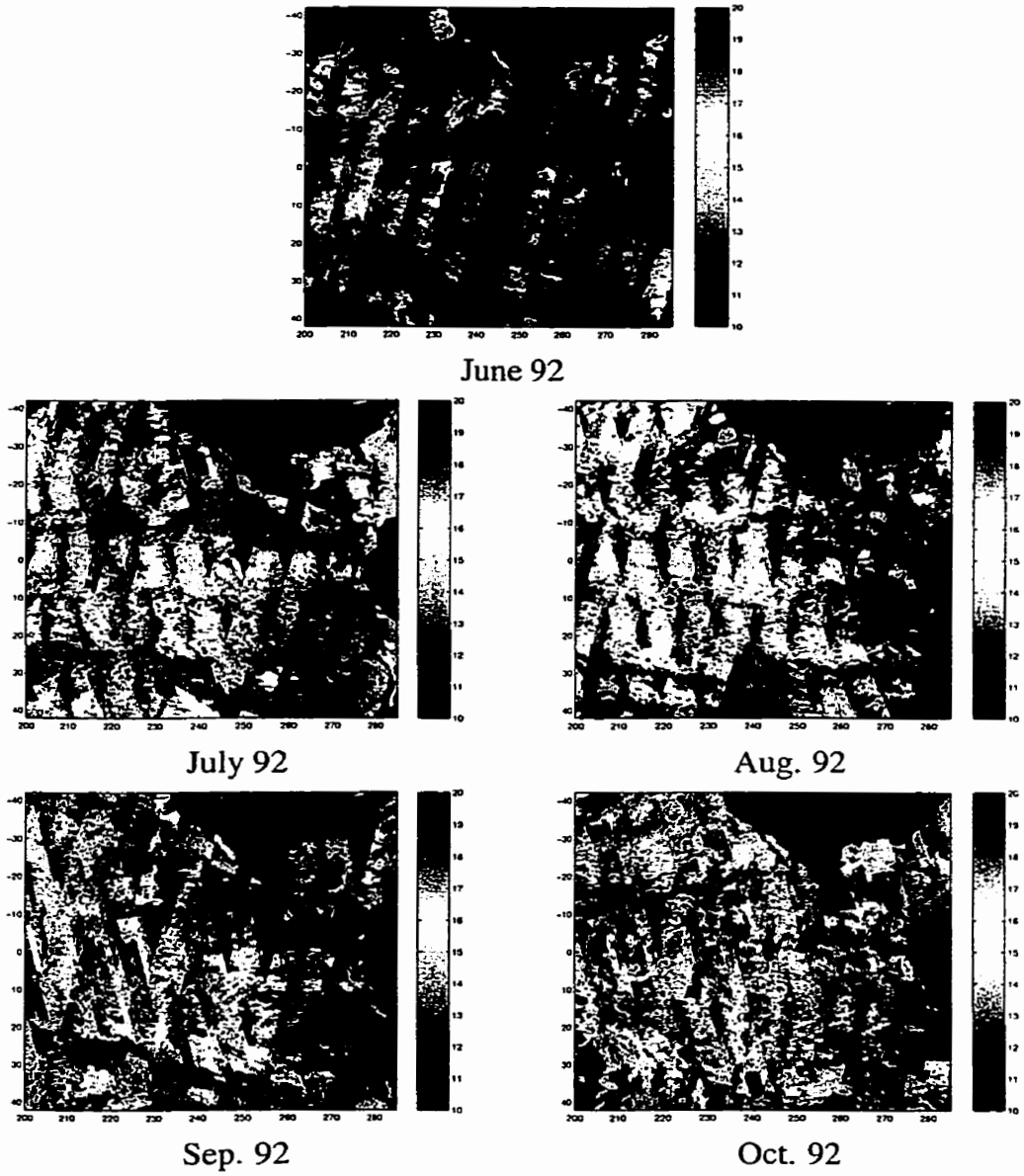


Figure 6.16: Corresponding estimation error std. dev. over a five months period

- The error standard deviations are reasonable in the sense that regions that are recently measured have a lower uncertainty than others, and there is a smooth transition, over time, from certainty to uncertainty.

- The presence of some sharp gradients in the estimates is the result of an inadequate dynamic model. Modeling the dynamics of the ocean surface temperature as a diffusion process does not consider any motion of the surface. As a result, we observe in the obtained estimates the existence of some hot patches in the vicinity of cold ones and vise versa. This suggests that better estimation results can be produced by having some sort of a motion model for the ocean surface.

Chapter 7

Thesis Summary

In this thesis we have presented a new approach to dynamic estimation that can efficiently deal with large-scale 2D processes. Large-scale dynamic estimation is important in many scientific problems. However, due to the high computational and storage demands for such large problems, the direct application of the Kalman filter becomes infeasible. This research is motivated by a real large-scale (512×512) estimation problem which is the estimation of the ocean surface temperature (OST) based on satellite observations. Dynamic estimation is suitable for such a problem because of the sparsity of the available data. Due the huge state vector size 2×10^5, the application of the Kalman filter becomes impossible.

One key aspect of the Kalman filter is that it computes estimates at any time based on all available observations up to that time without the need to store the previous observations. This is done by storing error covariance matrices and propagating them overtime. The size of the error covariance matrices grows with the size of the state vector. For example, for the OST problem, the filter needs to propagate error covariance matrices of size $10^5 \times 10^5$.

From the above, it becomes obvious that for large-scale 2-D dynamic estimation problems there exists a need to develop an *approximate* dynamic filter that emulates the

Kalman filter but with more efficient computational and storage demands.

Although there have been several research works that tried to deal with such issues for *relatively* large estimation problems, these studies mainly addressed the computational demands of the *update* step under some restricted and impractical assumptions:

- Fast Fourier transform (FFT) [33] methods require that the underlying process is spatially stationary.
- Iterative methods for solving the normal equations [48] require the full error matrices to be available in addition they do not explicitly provide the estimation error statistics.
- The sparse Kalman filter [16] exploits the sparsity of the filter matrices and uses polynomial approximation for matrix inversion. This method requires the availability of the error matrices in addition it requires that the matrix to be inverted to be diagonally dominant.
- Covariance extension methods [20, 40, 50] reduce the storage requirements for the error covariance matrices but are only developed for 1-D large estimation problems.
- Multiscale based dynamic estimation for 2D problems with reduced order tree states is only applicable for small size problems because it requires the covariance matrix of the error process at steady state to be available.

Our approach to this problem is based on using appropriate *parameterized* error models. Instead of storing large error covariance matrices, we store only the main diagonal elements and we use a parameterized model to represent the correlation structure. The main problem with the parameterized model, which we addressed in this research, is the imposed stationarity. As a fact, while the correlation structure of the diffusion process is stationary, the error process is not.

We began by studying the effect of the measurement update step on the error process correlation statistics. We empirically showed that in general the error statistics are spatially non-stationary and there is a smooth transition of the correlation structure from almost weak correlation at the measured positions to high correlation at places that have not been observed. By considering this observation, we introduced the notion of correlation length at any error process element in the 2D domain to encode the spatial non-stationarity in the error statistics.

In the prediction step, we presented a new approximate method to propagate the two components of the updated error covariance matrix: the updated estimation error variances and the non-stationary correlation structure. The method suits large-scale problems because it has almost linear computational and storage demands. This was presented in detail in Chapter 4.

For the update step, we adopted the multiscale estimator. Although computationally efficient and fast, it requires a stationary prior to model the underlying process. We showed that computing the update step based on stationary error models might lead to huge statistical inconsistencies. We presented a method to combine estimates based on several stationary priors. The method is capable of computing the associated estimation error variances.

We examined our method on several 2D examples under the context of diffusion dynamics. Based on the synthetic 2D diffusion experiments that are described in Chapter 6, the obtained estimation results are satisfactory and comparable to the optimal solution obtained by the Kalman filter. In addition, we have shown the real benefits of our approach in the context of a truly large-scale dynamic estimation problem related to remote sensing such as the OST problem. The ocean surface is modeled as a 2D diffusion process. We obtained approximate diffusion parameters that can fit the ocean spatial and temporal statistics. We applied our method to the sparse data of the ocean surface temperature and were able to obtain dense estimates for the ocean surface temperature based on a five month data set. We believe that the obtained results represent a good starting

point and require more investigation by scientists who are expert in the ocean science.

7.1 Suggestions for future work

As with most research efforts, this dissertation possibly raised as many questions as it solved. However, the ideas presented here should provide a strong foundation upon which to build and extend. In the following, we present some improvements to our current work which can be considered as future research directions.

For the update step, we adopted the multiscale estimator. We pointed out that for large-scale problems the accuracy and the quality of the produced estimates is highly dependent on the size of the internal tree states. As the length of the tree states increases better estimates are obtained. However, numerical errors might occur. A recently developed method by [89] can deal with large-scale 2D static estimation problems. It can also provides the necessary estimation error variances. The method is based on using the conjugate gradient algorithm. Although this method is based on using a stationary prior, it does not suffer from the numerical problems and the quality of the obtained estimates and the estimation error variances are almost comparable to the optimal solution obtained by solving the LSE exactly. It would be a good idea if one employed this method for solving the large-scale update step instead of the multiscale estimator. Since this method is almost exact, one can also study using it to compute the necessary interpolating weights for the state and for the estimation error variances.

When computing the optimal interpolation weights for the estimates and the associated estimation error variances we used the standard LSE. One can try to study the possibility of using the weighted LSE instead for some suitable weighing matrix W that can reflect the characteristics of the underlying estimation problem. As an example, one can study using a weighting matrix W that gives more weights to elements that have been measured.

For the real problem which is the estimation of the ocean surface temperature, we as-

sumed that the ocean surface is diffusive. We developed our prediction and update methods based on this assumption. The diffusion dynamics are simple and have a well-studied behaviour. The diffusion dynamics do not take care of any motion of the ocean surface which we do observe by displaying the dense estimates overtime. As a future work, one can try to estimate the ocean motion based on the sparse satellite measurements or based on the dense estimates obtained by our method. Motion estimation methods such as optic flow or block matching can be investigated. By combining both diffusion dynamics and motion estimates, the sharp gradients in the obtained temperature estimates can be reduced.

Other types of complex dynamics that suit the ocean such as the Rossby wave equation [97] can also be investigated. If one adopt the Rossby wave equation as the dynamic model then several issues have to investigated: (i) What correlation structure does the process have and (ii) if the process has a complicated correlation structure, what is the effect of imposing an exponential or Gaussian model on the obtained estimates and finally (iii) how can one do state and error predictions?

In general we suggest that any future extensions to this work and, in general, to large-scale 2D dynamic estimation problems should focus on increasing the quality of the obtained estimates in addition to decreasing the computational and storage demands.

Appendix A

Multiscale Algorithm

The multiscale process and the measurements models are given by:

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

$$y(s) = C(s)x(s) + v(s) \tag{A.2}$$

where $w(s)$ is $\mathcal{N}(0, I)$ a zero-mean and unit variance white noise process with normal distribution.

The measurements noise $v(s)$ is also white noise with zero-mean and covariance $R(s)$ but it is not multiple of the identity.

The process prior at the coarsest scale (i.e the root node)

$$x_o = x(0) \sim \mathcal{N}(0, P_o) \tag{A.3}$$

The upward model corresponding to the downward model given in (A.1)

$$x(s\bar{\gamma}) = F(s)x(s) + \bar{w}(s) \tag{A.4}$$

$$y(s) = C(s)x(s) + v(s) \tag{A.5}$$

where

$$F(s) = P(s\bar{\gamma})A^T(s)P(s)^{-1} \tag{A.6}$$

$$E[\bar{w}(s)\bar{w}(s)^T] = P(s\bar{\gamma}) - F(s)A(s)P(s\bar{\gamma}) = Q(s) \tag{A.7}$$

$P(s) = E[x(s)x^T(s)]$ is the prior covariance of state $x(s)$ and it can be computed by the Lyapunov recursive equation for the multiscale model

$$P(s) = A(s)P(s\bar{\gamma})A^T(s) + B(s)B^T(s) \tag{A.8}$$

Before giving the details of the algorithm, it is necessary to define some terms:

- $Y_s = \{y(\sigma)|\sigma$ is a descendant of $s\}$ is a set of measurements at all nodes below s excluding the measurement at node s
- $\hat{x}(\sigma|s) = E[x(\sigma)|\sigma \in Y_s \cup y(s)]$ is the best estimate of $x(\sigma)$ given measurements at node s and all nodes below s
- $\hat{x}(\sigma|s+) = E[x(\sigma)|\sigma \in Y_s]$ is the best estimate of $x(\sigma)$ given measurments at all nodes below s
- $\tilde{P}(\sigma|s) = \text{Cov}[x(\sigma) - \hat{x}(\sigma|s)]$
- $\tilde{P}(\sigma|s+) = \text{Cov}[x(\sigma) - \hat{x}(\sigma|s+)]$

The algorithm is consists of three main steps:

(A) Initialization

Each node s at the finest scale is assigned the following prior values:

$$\hat{x}(s|s+) = 0 \tag{A.9}$$

$$P(s|s+) = P(s) \tag{A.10}$$

(B) Upward Sweep

The upward sweep computes the best estimates of the state $x(s)$ at node s given measurements at or below node s. It consists of two steps at each scale

1. Update step:

$$\hat{x}(s|s) = \hat{x}(s|s+) + K(s)[y(s) - C(s)\hat{x}(s|s+)] \quad \text{(A.11)}$$

$$P(s|s) = [I - K(s)C(s)]P(s|s+) \quad \text{(A.12)}$$

$$K(s) = P(s|s+)C^T(s)[C(s)P(s|s+)C^T(s) + R(s)]^{-1} \quad \text{(A.13)}$$

2. Prediction step This step is applied to all nodes except the leaf nodes (which were initialized in the initialization step).

$$\hat{x}(s|s\alpha_i) = F(s\alpha_i)\hat{x}(s\alpha_i|s\alpha_i) \quad \text{(A.14)}$$

$$P(s|s\alpha_i) = F(s\alpha_i)P(s\alpha_i|s\alpha_i)F^T(s\alpha_i) + Q(s\alpha_i) \quad \text{(A.15)}$$

The above step will give the best estimate of $x(s)$ given all measurements at the offspring $s\alpha_i (i = 1 \ldots q)$. The q offsring based estimates of $x(s)$ are combined via the merge equations:

$$\hat{x}(s|s+) = P(s|s+)\sum_{i=1}^{q} P^{-1}(s|s\alpha_i)\hat{x}(s|s\alpha_i) \quad \text{(A.16)}$$

$$P(s|s+) = \left[(1-q)P(s)^{-1} + \sum_{i=1}^{q} P^{-1}(s|s\alpha_i)\right]^{-1} \quad \text{(A.17)}$$

(C) Downward Sweep The estimate at the root node $\hat{x}^s(0) = \hat{x}(0|0)$ is a smoothed one. The smoothed estimates of the remaining states are found by distributing the information back down the tree

$$\hat{x}^s = \hat{x}(s|s) + J(s)[\hat{x}^s(s\bar{\gamma}) - \hat{x}(s\bar{\gamma}|s)] \quad \text{(A.18)}$$

$$P^s(s) = P(s|s) + J(s)[P^s(s\bar{\gamma}) - P(s\bar{\gamma}|s)]J^T(s) \quad \text{(A.19)}$$

$$J(s) = P(s|s)F^T(s)P^{-1}(s\bar{\gamma}|s) \quad \text{(A.20)}$$

where

$\hat{x}^s(s)$ are the smoothed estimates

$P^s(s)$ are the corresponding estimation error covariances.

Bibliography

[1] Shin Ichi Aihara. "On adaptive boundary control for stochastic parabolic systems with unknown potential coefficient". *IEEE Trans. On Automatic Control*, Vol.42 No.3:350–363, 1997.

[2] B. Anderson and J. Moore. *Optimal Filtering*. Prentice-Hall, New York, 1979.

[3] Denise Angwin and Howard Kaufman. "Image restoration using reduced order models". *Signal Processing*, Vol. 16:21–28, 1989.

[4] Amir Asif and Jose Moura. "Data assimilation in large time-varying multidimensional fields". *IEEE Transactions on Image Processing*, Vol.8,No. 11:1593–1607, 1999.

[5] Mark Banham and Aggelos Katsaggelos. "Spatially adaptive wavelet-based multiscale image restoration". *IEEE Trans. on Image Processing*, Vol. 5 No. 4:619–634, 1996.

[6] Michel Basseville, Albert Benvensite, Kenneth Chou, Stuart Golden, Ramine Nikoukhan, and Alan Willsky. "Modeling and estimation of multiresolution stochastic processes". *IEEE Trans on Information Theory*, Vol. 38 No. 2:766–784, 1992.

[7] Julius Bendat and Allan Piersol. *Random Data: Analysis and Measurement Procedures*. Wiley and Sons, New York, 1971.

[8] Michael Berhold and David J. Hand. *Intelligent Data Analysis*. Springer, New York, 1997.

[9] Jan Biemond, Jelle Rieske, and Jan Gerbrands. "A fast Kalman filter for images degraded by both blur and noise". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 31 No. 5:1248–1255, 1983.

[10] Ake Bjorck. *Large Scale Matrix Problems*. North Holland, New York, 1981.

[11] Isabelle Blanchet and Claude Frankignoul. "A comparison of adaptive Kalman filters for a tropical pacific ocean model". *Monthly Weather Review*, Vol.125:40–57, 1997.

[12] Robert F. Brammer, Ralph P. Pass, and James V. White. "Bathymetric and oceanographic applications of Kalman filtering techniques". *IEEE Trans. on Automatic Control*, Vol.28 No. 3:363–370, 1983.

[13] A. Enis Cetin and Murat Teklap. "Robust reduced update Kalman filtering". *IEEE Trans. on Circuits and Systems*, Vol. 37 No. 1:155 – 156, 1990.

[14] T. Chin. On Kalman filter solution of space-time interpolation". *IEEE Transactions On Image Processing*, Vol.10, No. 4:663–666, 2001.

[15] T. Chin, W. Karl, and A. Willsky. "Sequential filtering for multi-frame visual reconstruction". *Signal Processing*, Vol.28:311–333, 1992.

[16] Toshio Chin. *Dynamical Estimation in Computational Vision*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1991.

[17] Kenneth Chou, Alan Willsky, and Albert Benvensite. "Multiscale recursive estimation, data fusion, and regularization". *IEEE Trans on Automatic Control*, Vol. 39 No. 3:464–477, 1994.

[18] Stuart Citrin and Mahmood Sadjadi. "A full-plane block Kalman filter for image restoration". *IEEE Trans. On Image Processing*, Vol. 1 No.4:488–494, 1992.

[19] A. Cracknell and W. Hayes. *Introduction to Remote Sensing*. Taylor and Francis, New York, 1991.

[20] A.P. Dempster. "Covariance selection". *Biometrics*, Vol.28:157–175, 1972.

[21] Haluk Derin and Patrick Kelly. "Discrete-index markov-type random processes". *Proceedings of the IEEE*, Vol. 77, No. 10:1485–1510, 1989.

[22] Sudhir Dikshit. "A recursive Kalman window approach to image restoration". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol.30 No.2:125–140, 1982.

[23] Renata Dmowska and Barry Saltzman. *Advances in Geophysics*. Academic Press, New York, 1991.

[24] N. Draper and H. Smith. *Applied Regression Analysis*. John Wiley, New York, 1966.

[25] Zeljko Durovic and Branko Kovacevic. "QQ-plot approach to robust Kalman filtering". *International Journal of Control*, Vol.61 No.4:836–857, 1995.

[26] Bernard C. Levy et. al. "Solution of linear estimation of 2D nearest neighbour model". *Proceedings of the IEEE*, Vol.78 No.4:627–641, 1990.

[27] John F. Galantowiez et. al. "Test of sequential data assimilation for retrieving profile soil moisture and temperature from observed l-band radiobrightness". *IEEE Trans. On Geoscience and Remote Sensing*, Vol.37 No.4:1860–1890, 1999.

[28] M. Srinath et. al. *Introduction to Statistical Signal Processing with Applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[29] S. Omatu et. al. "Estimation of nitrogen dioxide concentrations in the vicinity of a roadway by optimal filtering theory". *Automatica*, Vol.24 No.1:19–29, 1988.

[30] B. Everitt. *Introduction to Optimization Methods and their Application in Statistics*. Chapman and Hall, New York, 1987.

[31] Stanley Farlow. *Partial Differential Equations for Scientists and Engineers*. Dover Publications, Inc., New York, 1993.

[32] Paul Fieguth. *Application of multiscale estimation to large scale multidimensional and remote sensing problems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.

[33] Paul Fieguth. *SD774 - Modeling and Estimation - Lectures Notes*. 1998.

[34] Paul Fieguth, M. Allen, and M. Murray. "Hierarchical methods for global-scale estimation problems". *In Proceedings of Canadian Conference on Electrical Engineering, Waterloo, Canada*, pages 161–164, 1998.

[35] Paul Fieguth, F.Khellah, M. Allen, and M. Murray. "Large scale dynamic estimation of ocean surface temperature". *In Proceedings of IEEE 1999 International Geoscience and Remote Sensing Symposium*, Vol. 3:1826–1828, 1998.

[36] Paul Fieguth, F.Khellah, M. Allen, and M. Murray. "Data fusion of sea-surface data". *In Proceedings of IEEE 2000 International Geoscience and Remote Sensing Symposium*, 2000.

[37] Paul Fieguth, William Karl, and Alan Willsky. "Multiresolution optimal interpolation and statistical analysis of topex/poseidon satellite altimetry". *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 33 No. 2:280–292, 1995.

[38] Paul Fieguth, William Karl, and Alan Willsky. "Efficient multiresolution counterparts to variational methods for surface reconstruction". *Computer Vision and Image Understanding*, Vol. 70 No. 2:157–176, 1998.

[39] Paul Fieguth and Alan Willsky. "Fractal estimation using models on multiscale trees". *IEEE trans. on Signal Processing*, Vol. 44 No. 5:1297–1300, 1996.

[40] Austin Frakt. *Internal Autoregressive Processes, Stochastic Realization, and Covariance Extension.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1999.

[41] Arthur Gelb. *Applied Optimal Estimation.* MIT Press, Cambridge,MA, 1974.

[42] James Gentle. *Numerical Linear Algebra for Applications in Statistics.* Springer, New York, 1998.

[43] Alan George and Joseph W. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, London, 1991.

[44] Gene H. Golub and Charles F. *Matrix Computations.* The Johns Hopkins University Press, London, 1991.

[45] Mohinder Grewal and Angus Andrews. *Kalman Filtering Theory and Practice.* Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[46] Brown Grover and Patrick Hwang. *Introduction to Random Signals and Applied Kalman Filter.* Wiley, New York, 1997.

[47] Ali Habibi. "Two-dimensional Bayesian estimate of images". *Proceedings of the IEEE*, Vol.60 No.7:878–883, 1972.

[48] Wolfgung Hackbusch. *Iterative Solution of Large Sparse Systems of Equations.* Springer-Verlag, New York, 1994.

[49] Monson Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley and Sons, New York, 1996.

[50] Terrence Ho. *Multiscale Modeling and Estimation of Large Scale Dynamic Systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1998.

[51] Shang hua Teng. "Fast nested dissection for finite element meshes". *SIAM Journal on Matrix Analysis and Applications*, Vol. 18, No. 3:552–556, 1997.

[52] William Irving, Paul Fieguth, and Alan Willsky. "An overlapping tree approach to multiscale stochastic modeling and estimation". *IEEE Trans. on Image Processing*, Vol. 6 No. 11:1517–1529, 1997.

[53] William Irving, William Karl, and Alan Willsky. "A theory for multiscale stochastic realization". *In Proceedings of the 33rd Conference on Decision and Control, Lake Buena Vista*, pages 655–662, 1994.

[54] Michael Isard and Andrew Blake. "Condensation-conditional density propagation for visual tracking". *International Journal of Computer Vision*, Vol.29 No.1:5–28, 1998.

[55] Anil K. Jain. "Advances in mathematical models for image processing". *Proceedings of the IEEE*, Vol.69 No.5:502–525, 1981.

[56] Anil K. Jain and Jaswant R. Jain. "Partial differential equations and finite difference methods in image processing-part ii: Image restoration". *IEEE Trans. On Automatic Control*, Vol.23 No.5:817–835, 1978.

[57] Anil K. Jain and Joachim Jasidek. "Fast fourier transform algorithms for linear estimation, smoothing, and Riccati equations". *IEEE Trans. On Acoustics, Speech, and Signal Processing*, Vol.31 No.6:1435–1446, 1983.

[58] Jaswant Jain and Anil Jain. "Displacement measurement and its application in interframe image coding". *IEEE Trans. on Communications*, Vol.29 No.12:1799–1808, 1981.

[59] J.W.Woods and C. Radewan. "Kalman filtering in two dimensions". *IEEE Trans. Inform. Theory*, Vol. IT-23:437–481, 1977.

[60] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation theory*. Prentice Hall, New Jersey, 1993.

[61] Marc Lavielle. "2-D Bayesian deconvolution". *Geophysics*, Vol.56 No.12:2008–2018, 1991.

[62] Edward Lee and Simon Haykin. "Parallel implementation of the extended square-root covariance filter for tracking applications". *IEEE on Parallel and Distributed Systems*, Vol.4 No. 4:446–465, 1993.

[63] C. Leondes. *Advances in Control Systems, Theory and Applications*. Academic Press, New York, 1966.

[64] Hanoch Lev-Ari, Sydney R. Porter, and Thomas Kailath. "Multidimensional maximum-entropy covariance extension". *IEEE Trans. on Information Theory*, Vol. 350 No. 3:497–508, 1989.

[65] Frank L. Lewis. *Optimal Estimation with an Introduction to Stochastic Control Theory*. John Wiley and Sons, New York, 1986.

[66] Mark Luettgen. *Image Processing with Multiscale Stochastic Models*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1993.

[67] Mark Luettgen and Alan Willsky. "Multiscale smoothing error models". *IEEE trans. on Automatic Control*, Vol. 40 No. 1:173–175, 1995.

[68] Larry Matties, Takeo Kanade, and Richard Szeliski. "Kalman filter-based algorithms for estimating depth from image sequences". *International Journal of Computer Vision*, Vol.3:209–236, 1989.

[69] Peter Maybeck. *Stochastic Models, Estimation,and Control Vol.1*. Academic Press, New York, 1979.

[70] Richard J. Meinhold and Nozer D. Singpurwalla. "The Kalman filter model: Motivation and applications". *The American Statistician*, Vol.37 No.2:123–127, 1983.

[71] Jerry Mendel. *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Prentice Hall, New Jersey, 1995.

[72] Dimitris Menemenlis, Paul Fieguth, and Alan Willsky. "Adaption of fast optimal interpolation algorithm to the mapping of oceanographic data". *Journal of Geophysical Research*, Vol. 102 No. C5:10573–10583, 1997.

[73] Robert N. Miller. "Toward the application of the Kalman filter to regional open ocean modeling". *Journal of Physical Oceanography*, Vol.16:72–86, 1985.

[74] Dwight Mix. *Random Signal Processing*. Prentice Hall, New Jersey, 1995.

[75] M. Murry and M. Allen. "Actual and potential information in dual-view radiometer observations of sea surface temperature from atsr". *Journal of Geophysical Research*, Vol. 103, No. C4:8153–8165, 1998.

[76] C. Mutlow and A. Zavody. "Sea surface temperature measurements by the along -track scanning radiometer on the ers1 satellite: Early results". *Journal of Geophysical Research*, Vol. 99, No. C11:22575–22588, 1994.

[77] Gerado Noriega and Subbarayan Pasupathy. "Application of Kalman filtering to real-time processing of geophysical data". *IEEE Trans. on Geoscience and Remote Sensing*, Vol.30 No.5:897–883, 1992.

[78] Ricardo Olea. *Geostatistics for Engineers and Earth Scientists.* Kluwer Academic Publishers, London, 1999.

[79] Steven Powell and Leonard Silerman. "Modeling of two-dimensional covariance functions with application to image processing". *IEEE Trans. on Automatic Control*, Vol.19 No.1:8–12, 1974.

[80] Andre Preumont. *Random Vibration and Spectral Analysis.* Kluwer Academic Publishers, Netherlands, 1990.

[81] A. Rangarajan, R. Chellappa, and V. Zhou. "A model-based approach for filtering and edge detection in noisy images". *IEEE Trans. on Circuits and Systems*, Vol.37 No.1:140–144, 1990.

[82] Radhakrishna Rao and Helge Toutenburg. *Linear Models, Least Squares and Alternatives.* Springer, New York, 1999.

[83] Alvin Rencher. *Linear Models in Statistics.* John Wiley, New York, 2000.

[84] Ian B. Rhodes. "A tutorial introduction to estimation and filtering". *IEEE Trans. on Automatic Control*, Vol.16 No.6:688–702, 1990.

[85] Kurt Rolle. *Heat and Mass Transfer.* Prentice Hall, New Jersey, 2000.

[86] A. Rosenfeld. *Multiresolution Image Processing and Analysis.* Springer-Verlag, New York, 1984.

[87] Reuven Rubinstein. *Simulation and The Monte Carlo Method.* Wiley, New York, 1981.

[88] Michael Schneider, Paul Fieguth, William Karl, and Alan Willsky. "Multiscale methods for segmentation and reconstruction of signals and images". *IEEE Trans. on Image Processing*, Vol. 9 No. 3:456–468, 2000.

[89] Michael Schneider and Alan Willsky. "Krylov subspace estimation". *Submitted to SIAM.*

[90] Richard R. Schultz and Robert L. Stevenson. "Stochastic modeling and estimation of multispectral image data". *IEEE Trans. On Image Processing*, Vol.4 No. 8:1109–1113, 1995.

[91] Fred Schweppe. *Uncertain Dynamic Systems.* Prentice Hall, New Jersey, 1973.

[92] K. Shanmugan. *Random Signals: Detection, Estimation and Data Analysis.* John Wiley and Sons, New York, 1988.

[93] K.W Simon and A.R. Stubberud. "Reduced order Kalman filter". *International Journal of Control*, Vol.10 No.5:501–509, 1969.

[94] Gilbert Strang. *Linear Algebra and its Applications.* Academic Press, New York, 1980.

[95] Bindinganavile Suresh and B. Shenoi. "New results in two-dimensional Kalman filtering with application to image restoration". *IEEE Trans. On Circuits and Systems*, Vol.28 No. 4:307–318, 1981.

[96] Demetri Terzopoulos. "Image analysis using multigrid relaxation methods". *IEEE Trans. On PAMI*, Vol. PAMI-8, No.2:129–138, 1986.

[97] David Tolmazin. *Elements of Dynamic Oceanography.* Boton, Allen and Unwin, 1985.

[98] E. Twizell. *Computational Methods for Partial Differential Equations.* Wiley, New York, 1984.

[99] Han Wackernagel. *Multivariate Geostatistics.* Springer, New York, 1998.

[100] Wen-Rong and Amlan Kundu. "Image estimation using fast modified reduced update Kalman filter". *IEEE Trans. On Signal Processing*, Vol.40 No.4:915–926, 1992.

[101] Alan Willsky and Gerogry Wornell. *Stochastic Processes, Detection and Estimation - Lectures Notes*. 1994.

[102] M. Woolfson and G. Pert. *Introduction to Computer Simulation*. Oxford University Press, New York, 1999.

[103] Gregory Wornell. "A Karhunen-Loeve-like expansion for 1/f processes via wavelets". *IEEE Transactions On Information Theory*, Vol. 36, No. 4:859–861, 1990.

[104] Zhe Wu. "Multidimensional state-space model Kalman filtering with application to image restoration". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 33 No. 6:1576–1592, 1985.

[105] Shan Yu and Gerard Giraudon. "Toward robust analysis of satellite images using map information-application to urban area detection". *IEEE Trans. On Geoscience and Remote Sensing*, Vol. 37 No. 4:1925–1938, 1999.