MULTI-RESOLUTION IMAGE FUSION USING MULTI-SCALE ESTIMATION FRAMEWORK

By

HOJIN JHEE

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2010

© 2010 Hojin Jhee

To my parents and friends

ACKNOWLEDGMENTS

I wish to sincerely thank my academic advisor, Dr. Fred Taylor, for providing me a chance to keep pursuing my graduate studies at University of Florida. I deeply appreciate his notable support, friendly advice, and consideration in guiding for completing this dissertation. I am grateful to all my supervisory committee members: Dr. Herman Lam, Dr. Janise McNair and Dr. Douglas Cenzer sharing their valuable time for great comments and interests by many ways during my Ph.D. period. Also, I need to send deep thanks to Dr. Clint Slatton who served as my former academic advisor. He greatly inspired my Ph.D. research. He was a respectable researcher, instructor and sometimes, nice friend during the periods when I was under his advice. I will always keep him and his family in my thoughts and pray for his peaceful rest.

Special thanks go to all my colleagues for their helps, encouragements and nice friendships during my joyful experiences at University of Florida.

Lastly, I would like to express utmost respect and appreciation to my parents who have patiently and endlessly offered support, inspiration and motivation for achieving my academic dream. I owe much them all.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS 4

LIST OF TABLES 7

LIST OF FIGURES 8

LIST OF ABBREVIATIONS 11

ABSTRACT 13

1 INTRODUCTION 15

1.1 Background 15
1.2 Multi-Scale Image Fusion 16
1.3 Problem Statement and Organization 18

2 OVERVIEW OF MULTI-SCALE ESTIMATION FRAMEWORK 21

2.1 Introduction 21
2.2 Multi-Scale Data Representation 22
2.2.1 State Space Models on q^{th} Order Tree 22
2.2.2 Markov Property of Multi-Scale Process 23
2.3 Conclusions 25

3 DATA FUSION BY MULTI-SCALE KALMAN SMOOTHING ALGORITHM 27

3.1 Introduction 27
3.2 Preliminaries 28
3.3 The Kalman Filter 29
3.3.1 Linear State Model 29
3.3.2 Kalman Filtering Algorithm 30
3.3.3 Kalman Smoothing Algorithm 30
3.4 Multi-Scale Kalman Smoothing (MKS) Algorithm 32
3.5 Conclusions 36

4 APPLICATIONS 39

4.1 Introduction 39
4.2 Reduced-Complexity MKS (RC-MKS) 40
4.2.1 Motivations 40
4.2.2 Sparse MKS Implementation 41
4.2.3 Image Fusion Results Using RC-MKS 42
4.2.4 Conclusions 43
4.3 Vector-Valued MKS 44

4.3.1 Motivations 44
4.3.2 Standard MKS Using Vector-Valued Measurements 44
4.3.3 The Information Form of Kalman Filter 45
4.3.4 Efficient Measurement Updates Methods 45
4.3.4.1 Serial measurement updates method 45
4.3.4.2 Selective measurement updates method 48
4.3.5 Image Fusion Results Using Vector-Valued MKS 48
4.3.6 Conclusions 50

5 QUADTREE IMAGE MODEL AND BLOCKY ARTIFACT 55

5.1 Introduction 55
5.2 Example: Comparisons of Covariance Structures among Three Different Graphical Models 55
5.2.1 Introduction 55
5.2.2 Estimation of Gaussian Process 57
5.2.3 Multi-Scale Modeling Using Pyramidal Tree 57
5.2.3.1 Prior model of pyramidal tree 58
5.2.3.2 Intra-scale structure (For Θ_t) 58
5.2.3.3 Inter-scale structure (For Θ_s) 59
5.2.4 Comparisons of Correlation Decays (Mono-Scale, Dyadic Tree and Pyramidal Tree) 60
5.3 Conclusions 61

6 IMAGE FUSION USING SINGLE FRAME SUPER RESOLUTION 66

6.1 Introduction 66
6.2 Super Resolution Method 69
6.3 Super Resolution Multi-Scale Kaman Smoothing Algorithm (SR-MKS) 70
6.3.1 Summary of Super Resolution Algorithm 71
6.3.2 Feature Representation 74
6.3.3 Super Resolution Multi-Scale Kalman Smoothing (SR-MKS) 75
6.4 Simulations 75
6.5 Conclusions 76

7 CONCLUSIONS AND FUTUREWORKS 84

7.1 Conclusions 84
7.2 Future Works 87

LIST OF REFERENCES 89

BIOGRAPHICAL SKETCH 94

LIST OF TABLES

Table page

4-1 Comparison of memory storage using RC-MKS and standard MKS. (Mb:Mega bytes, Gb:Giga bytes) 53

4-2 ERS and TOPSAR system parameters and σ_h (RMS error variance) 53

4-3 Comparison of error performance using single TOPSAR MKS and the proposed method. (All elevation units are in meters.) 54

6-1 Mean square errors of image fusion results in Figure 6-6 (in meters) 83

LIST OF FIGURES

Figure page

1-1 An example of image fusion using multi-scale estimation framework. DEMs of ERS-1 (20m spacing at 9th scale) and TOPSAR (5m spacing at 11th scale) are optimally fused together by employing Multi-scale Kalman Smoothing (MKS) algorithm. We have to notice that TOPSAR at 11th scale suffers severe data dropouts (represented by white). Also, the final fused estimated at 11th scale is shown. All elevation units are in meters. 20

2-1 Multi-scale tree structures. A) 1-D dyadic structure, and B) 2-D quadtree structure. The root node corresponds to the coarsest scale while the leaf nodes comprise the finest scale. 26

2-2 The first four levels of 2nd order tree structure (dyadic tree). The parent node of s is represented by Bs and two offspring are denoted by $s\alpha_1$ and $s\alpha_2$. 26

2-3 Markov property of dyadic tree structure. Conditioned on node s, the nodes in the corresponding 3 subtrees of nodes extending away from s are uncorrelated. (Each subset of nodes is represented by yellow for ψ_{s1}, green for ψ_{s2} and grey for ψ_{s3}, respectively) 26

3-1 Recursion diagram of a Kalman filter..... 37

3-2 Two sweep steps of time-series Rauch-Tung-Striebel algorithm. 1) Forward sweep (Kalman filtering): Optimal inference on the hidden variables $x(t)$ given a collection of past and present measurements $y_0, y_{1,\ldots} y_t$, ,and 2) Backward sweep (Smoothing): Recursively computes the quantities of estimates given all measurements $y_0, y_{1,\ldots} y_T$. 37

3-3 The MKS algorithm. A) Upward sweep (Kalman filtering) computes (3.9) ~ (3.12) from fine to coarse scale, and B) Downward sweep (Kalman smoothing) computes (3.13) from coarse to fine scale. 38

3-4 An example of image fusion by employing MKS algorithm. A) ERS-1 elevation data spaced by 20m, B) TOPSAR elevation data spaced by 5m, C) Fused estimate at the finest scale (11th scale), and D) RMS error of final estimate. All elevation units are in meters. 38

4-1 The finest resolution measurement is sparse relative to the coarse-scale measurement. The highly sparse fine-scale image is at 13th scale (5m spacing), and dense coarse-scale image is at 9th scale (80m spacing) on quadtree. The data voids are represented by grey. All elevation units are in meter. 51

4-2 Set of subtrees for which finest-scale data are available. (each set of "valid" subtrees are represented by yellow) 51

4-3 Perspective views of topographic and bathymetric elevations fused using the RC-MKS method. The coverage of area is 40 Km $\times$ 40 Km. A) Fused estimates (elevations) at 13th scale, and B) its RMS error. All elevation units are in meters 52

4-4 Image data sets to be fused. A) ERS-1 elevation data spaced by 20m, B) 1st TOPSAR elevation data spaced by 5m, C) 2nd TOPSAR elevation data spaced by 5m, and D) 3rd TOPSAR elevation data spaced by 5m. Dark blue area at each TOPSAR image represents data voids. All elevation units are in meters. 52

4-5 Fusion results using vector-valued MKS. A) Fused estimate, and B) its height uncertainty using serial measurement update method. All elevation units are in meters 53

5-1 An example of blocky artifact. In this example, two images are being fused by MKS algorithm in Chapter 3. At final estimate, we can notice that there are blocky regions shown, and these appear at the pixel locations where fine-scale image pixel values are not available. (Data missing is represented by white at fine-scale image). All elevation units are in meter. 62

5-2 Three process models used in this example. A) mono-scale Markov chain, B) dyadic tree structure, and C) pyramidal tree structure. The edges (or connections) between nodes represent the statistical dependency between a node and its neighborhood ones. 62

5-3 Inter-scale (Θ_s) and intra-scale (Θ_t) prior models in pyramidal tree structure. .. 63

5-4 Pyramidal tree structure (1-D case). Penalizing parameters α_m and β_m of each scale are represented on pyramidal tree structure. 63

5-5 A) 1-D mono-scale Markov chain structure (64 variables), B) Dyadic tree structure (64 variables at finest scale), and C) pyramidal tree structure (64 variables at finest scale). 64

5-6 Obtained covariance matrices ($\Theta^{-1} = P$) in (5.6) using A) mono-scale Markov chain, B) dyadic tree, and C) Pyramidal tree. 65

5-7 Correlation decay curves (all values are in log scale) of three different structures at the finest scale (1-D case). Parameters used in (1) Mono-scale: $\alpha_M = 1$, $\beta_m = \mathrm{N/A}$, (2) Dyadic tree: $\alpha_m = \mathrm{N/A}$, $\beta_m = [1\ 1\ 1\ 1\ 1\ 1\ 1]$, and (3) Pyramidal tree: $\alpha_m = [1\ 1\ 1\ 1\ 1\ 1\ 1]$ and $\beta_m = [1\ 1\ 1\ 1\ 1\ 1]$ 65

6-1 An example of nearest neighborhood searching algorithm applied to 5×5 low-resolution patch for 4 times magnification: A) input low resolution patch, B) 4 nearest neighbor patches extracted from the training images, C) reconstructed input patch by weighted combination of image patches in (B). (Weighting vector used: Γ_n={ 0.4787, 0.3110, 0.4518,-0.2415}). Darker red represents higher elevation. 78

6-2 A) Corresponding high resolution patch of Figure 6-1 (A), B) 4 high resolution patches corresponding to low resolution patches in (A), C) estimated high resolution patch constructed by (B). (Weighting vector used: Γ_n={ 0.4787, 0.3110, 0.4518,-0.2415}) Darker red represents higher elevation. 79

6-3 8 sets of training images used for the results in Figure 6-1 and Figure 6-2. Only high resolution training images are shown. Each of corresponding low resolution training images is obtained by down sampling by order of 4. 80

6-4 A 5×5 local neighborhood in the low-resolution image for computing the first-order and second-order gradients of the pixel at the center with elevation value x_{33} 81

6-5 The input image sets to the fusion algorithm. A) Ground truth, B) low resolution image (256×256), C) sparse high resolution image (1024×1024: data void regions are represented as white), D) Super resolved image of smoothed estimate at coarse scale (at 9th scale). All elevation units are in meters. 81

6-6 Comparisons of fused estimates at the finest scale (11th scale). A) using standard MKS introduced in Chapter 3, B) using proposed SR-MKS in Chapter 6, C) zoomed fusion results of data void areas (areas circled by red in (A)). 1st row: zoomed area at 1, 2nd row: zoomed area at 2, and 3rd row zoomed area at 3. For each row of result (left) fine-scale ground truth, (center) fused by MKS in Chapter 3, and (right) fused by proposed method (SR-MKS). All elevation units are in meters..... 82

LIST OF ABBREVIATIONS

DEM	Digital Elevation Model
ERS-1	European Remote Sensing satellite-1
fBm	fractional Brownian motion
Gb	Giga bytes
IHRC	International Hurricane Research Center
InSAR	Interferometric Synthetic Aperture Radar
JPL	Jet Propulsion Laboratory
KNN	K Nearest Neighborhood
LiDAR	Light Detection and Ranging Radar
LMMSE	Linear Minimum Mean Square Error
MAP	Maximum A Posteriori
Mb	Mega bytes
MKS	Multi-scale Kalman Smoothing
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MRFs	Markov Random Fields
NASA	National Aeronautics and Space Administration
NGDC	National Geophysical Data Center
NOAA	National Oceanic and Atmospheric Administration
PacRim	Pacific Rim
PSF	Point Spread Function
RMS	Root Mean Square
SR	Super Resolution
SR-MKS	Super Resolution Multi-scale Kalman Smoothing

RC-MKS	Reduced Complexity Multi-Scale Kalman Smoothing
RTS	Rauch Tung Striebel
TOPSAR	Topographic Synthetic Aperture Radar
1-D	One Dimensional
2-D	Two Dimensional

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

MULTI-RESOLUTION IMAGE FUSION USING MULTI-SCALE ESTIMATION FRAMEWORK

By

Hojin Jhee

May 2010

Chair: Fred J. Taylor
Major: Electrical and Computer Engineering

Recently, we have been experiencing remarkable advance in remote sensing technology and it allows us to capture large classes of natural processes and phenomenon at different resolutions with confident levels of qualities. For example, data acquisition over specific topographic environment by employing high spatial resolution sensor is extremely useful in monitoring detail physical and biological processes of the Earth's surface. As data acquisition techniques become sophisticated using more accurate sensing devices, the demands for processing obtained data sets are more diverse and complex.

In this dissertation, we develop data fusion methods to process image sets obtained by heterogeneous sources at different resolutions. This fusion scenario is based upon the idea that tries to combine image sets differing resolutions by employing robust and efficient signal processing scheme like multi-scale estimation framework. Since the data collection processes have been performed by different methods and for different purposes, merging process is not a trivial task. Despite of this technical difficulty, real world remote sensing applications require information that is insufficient to be interpreted by a single sensor measurement. Since individual sensor employed is

operated in certain acquisition geometries (e.g. altitude, distance or viewing angle), this turns disparate coverage and accuracy characteristics on obtained data. A number of attempts have been made to combine data from different sensors, but existing methods are often empirical based [Sorenson, 1970]. Successful data fusion becomes especially difficult when the sensors involved have significantly different acquisition methods, wavelengths, and resolutions.

To overcome this difficulty, this study shows the efforts to build robust image processing framework for combining (fusing) complementary multi-resolution image data sets. The contributions of this work are summarized as: (1) constructing statistically optimal fusion method utilizing multi-scale estimation framework such that one can efficiently obtain fused image estimate and its confident measure (uncertainty) at desired resolution, (2) developing new multi-scale fusion techniques to find solutions for computationally challenging fusion situations, and (3) extending multi-scale estimation techniques to generate both visually and analytically improved fused image at the finest resolution by mitigating pixel blocky artifacts commonly arising when the resolutions of the images being fused are differed by large orders of magnitudes and image pixel voids at fine-scale are severe.

CHAPTER 1
INTRODUCTION

1.1 Background

Merging or fusing image measurements captured by disparate sensing techniques has been a challenging image processing problem. Because of the geometric constrains on individual sensor platform such as altitude, distance or viewing angle, the acquired image from each sensor shows different characteristics in coverage and accuracy [Hall, 1997]. Despite of these characteristic diversities, real world remote sensing applications require information that is difficult to obtain with a single sensor measurement [Desmet, 1996]. For instance, a study of monitoring natural phenomenon at coast line for prediction of inland flood surge or sediment transport associated with near-shore area typically requires gridded Digital Elevation Model (DEM) which can cover large areas with high resolution and small elevation errors [NOAA; Zhang, 2003]. Since terrain data formed by DEM provide basic information for understanding and predicting natural systems in coast line, generation of accurate elevation model is the most fundamental but important work [Zebker, 1994]. DEM data acquired from space-borne based sensor generally provide good measures of topographical shapes over large areas, but locally the spatial resolution is insufficient for delineating details of coast line [Komar, 1998]. To achieve finer map, simple re-sampling (interpolation) process on obtained DEM can be applied, but increased cell size does not actually enhance spatial resolution of original data since no additional information is brought [Gesch, 2001]. On the other hand, a DEM created from airborne based platform such as Laser altimeter sensors (e.g. LiDAR (Light Detection and Ranging)) [IHRC, 2004] contains detail information about shore line shape with sub-meter height accuracy, because it typically

scans laser pulses through a small angle and footprints with lower altitude than space shuttle ones. These scanning geometries allow us to acquire very high resolution terrain data [Carter, 2001]. Yet, since imaging swath is normally smaller than that of space shuttle based sensor, acquired DEM from this type of platform commonly suffers missed elevation values or data voids.

Thus, the motivation of multi-resolution image fusion arises when we are requested to obtain a seamless integration of image data from multiple resolutions such that fused result has both extensive coverage and locally high-resolution details. By taking advantages of the availability of various data sets, a method has to be proposed for the synthesis of a fusion process. In this work, a problem of combining sparse and dissimilar data types at multiple resolutions will be solved by introducing a theoretically rigid method that produces globally optimal estimate at desired resolutions. In many applications, the fused estimate at the achievable highest resolution is much desirable. This image fusion goal will be achieved by employing multi-scale estimation framework and its results remain nearly optimal in the mean squared error sense.

1.2 Multi-Scale Image Fusion

A hierarchical signal modeling has been an attractive approach in signal and image processing areas for last decades [Willsky, 2002; Gonzalez, 2002]. This has been motivated by the needs to (1) develop stochastic model that is able to capture multi-scale characters of natural processes, and (2) process multiple measurements having different resolutions [Gallant, 2006]. Originally, the multi-scale data modeling is based on the idea that one can transform spatial data models to fine-to-coarse (or vice versa) method that directly models processes of interests on multi-scale data structures such as dyadic or quadtrees [Daniel, 1997]. The additional nodes within structure

correspond to coarser representations of the original nodes at the finer scale. The work proposed here uses quadtree structure to combine multiple image data differed by resolutions. A multi-scale estimation framework can be widely applicable in many remote sensing applications because multiple data sets of different resolutions are often needed to characterize the processes under study [Choi, 2007]. Chou et al., introduced a multi-scale linear estimation method based on the Kalman filter [Chou & Nikoukhah, 1994]. Kumar [Kumar, 1999] applied it to the problem of combining soil moisture data at different resolutions, while other investigators used the method for data interpolation and fusion [Fieguth & Karl, 1995; Daniel, 1997]. Chou, et al. and Fieguth, et al. derived a recursive estimator consisting of a Multi-scale Kalman Smoothing (MKS) constructed on a Markov tree data structure that accommodates multi-sensor measurements of differing resolutions [Chou & Benveniste, 1994; Fieguth & Irving, 1995]. The fused estimate with MKS allows the fusion of state variables that are not directly observed.

Figure 1-1 illustrates an example of image fusion (pixel level) using MKS algorithm for a pair of multi-resolution image sets measured from The Finke River in central Australia. [Slatton, 2000]. A dense low resolution image at 9th level (scale) of image pyramid is acquired by space-borne ERS-1 (European Remote Sensing satellite) processed to a spatial resolution of 20 meters, and high resolution one at 11th level is obtained by air-borne TOPSAR (Topographic Synthetic Aperture Radar) gridded to resolution of 5 meters [Slatton, 2000]. In this example, we can notice that the sensing range of TOPSAR is much limited comparing with coverage of the ERS-1, so it results in TOPSAR image yielding data dropouts or data voids. The MKS data fusion algorithm tries to fill these data dropouts by associating two-way estimation scheme on quadtree

structure. More details about multi-scale model and fine-to-coarse fusion algorithm used in this example will be covered in Chapter 2 and Chapter 3.

1.3 Problem Statement and Organization

The major objective of this dissertation is to demonstrate efforts for solving multi-resolution image fusion problems commonly encountered in remote sensing applications. For this goal, we employed multi-scale estimation approaches to construct standard (Chapter 3) and computationally favorable MKS algorithms (Chapter 4) which produce statistically optimal image fusion results. The main focus, in particular, is to complete sparsely populated fine-scale image by incorporating densely spaced coarse-scale image set on hierarchical data structure. While proposed methods, in general, provide outstanding performance in terms of accuracy and efficiency, such tree-based method is often limited to be applied in certain applications, for example, when smoothness of estimated result is essentially needed for meaningful computations of gradients and curvatures [Fieguth & Irving, 1995]. This is caused by the limitation that MKS algorithm usually exhibits blocky artifact at regions of the fused image where only coarse data are available. More precisely, due to the blocky covariance structure introduced by quadtree, the computed estimate suffers distracting pixel blocky artifact at locations where fine-scale pixels are void. Although blockiness on fused estimate can be mitigated by simple post-processing (e.g., the application of a lowpass filter, or the averaging of multiple), this often deteriorates the resolution of fine-scale details because the obtained smoothness is achieved by spatial blurring. Recently, large classes of flexible tree structures, such as dynamic or loopy tree, have been proposed to handle pixel blockiness, but their computational costs are still too expensive for large-scale image problems. [Murphy, 1999; Murphy, 2002; Adams, 2001; Todorovic, 2007]. Since

our goal for this work has been inspired by developing computationally efficient algorithm in optimal sense, the choice of quadtree is sill reasonable.

The following statements claim the purposes of this dissertation.

In this dissertation, effective multi-resolution image fusion framework will be proposed to merge multiple image sets having different resolutions. Learning based image resolution enhancement approach will be introduced and applied on proposed fusion methods to reduce blocky artifacts exhibited by quadtree image model. This new fusion scheme will successfully infer missed detail information at data void region in fine-scale image, so that the blockiness on fused estimate can be suppressed.

The organization of dissertation is as follows: In Chapter 2, we summarize multi-scale data representation which plays an important role in our fusion study. Chapter 3 introduces a multi-scale estimation approach based on Kalman filter. Multi-scale Kalman smoothing (MKS) algorithm is a generalized Rauch-Tung-Striebel (RTS) [Brown, 1997] estimator implemented on quadtree. At each node in the tree, the MKS optimally (in a least squared error sense) blends a stochastic multi-scale model with the available measurements according to a Kalman gain. In Chapter 4, we investigate computationally more challenging fusion problems by introducing new fusion algorithms which satisfy efficiency and accuracy on estimation results. Chapter 5 discusses about blocky artifact exhibited in MKS algorithm. To overcome the problem of blockiness, a new class of fusion scheme will be derived in Chapter 6 by employing learning-based image super resolution technique. Finally, Chapter 7 gives conclusion of this dissertation, together with suggestions for future works.

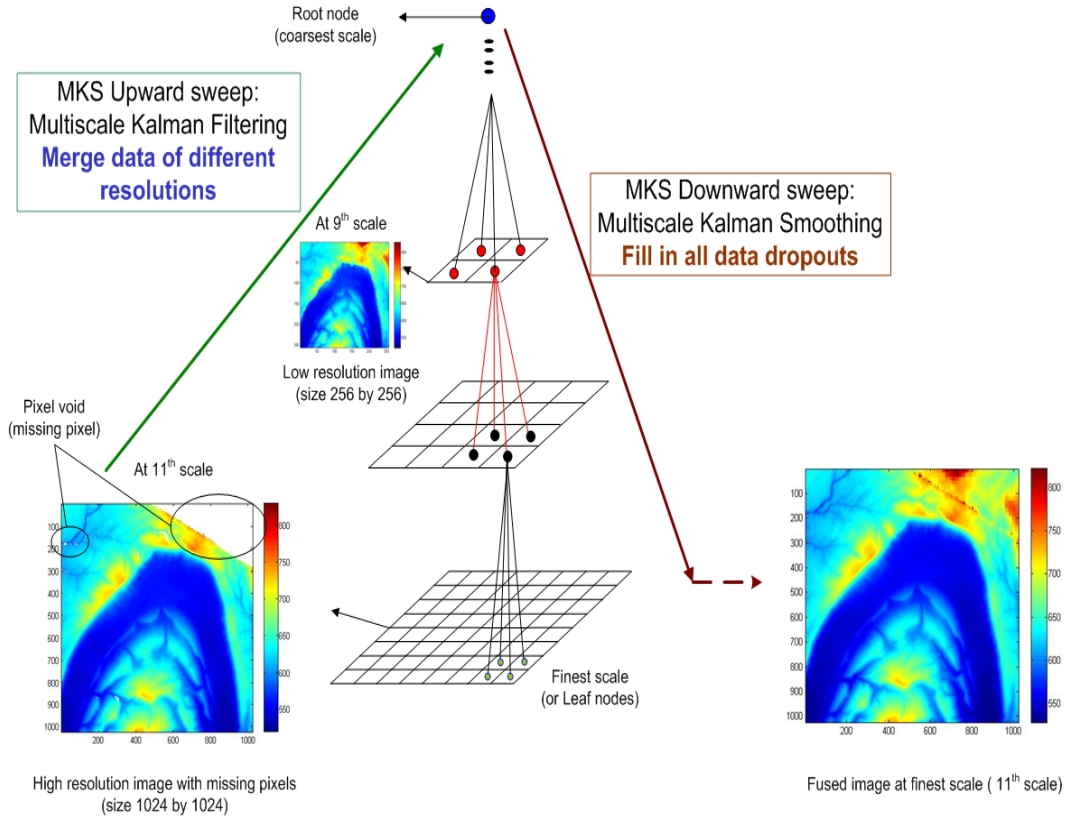


Figure 1-1. An example of image fusion using multi-scale estimation framework. DEMs of ERS-1 (20m spacing at 9th scale) and TOPSAR (5m spacing at 11th scale) are optimally fused together by employing Multi-scale Kalman Smoothing (MKS) algorithm. We have to notice that TOPSAR at 11th scale suffers severe data dropouts (represented by white). Also, the final fused estimated at 11th scale is shown. All elevation units are in meters.

CHAPTER 2
OVERVIEW OF MULTI-SCALE ESTIMATION FRAMEWORK

2.1 Introduction

A modeling of Gaussian processes indexed by the nodes of tree structure was introduced in [Chou & Nikoukhah, 1994]. For last decades, Markov random fields (MRFs) have been popularly used for image analysis [Laferte, 2000]. While MRFs provide a rich structure for multidimensional modeling, they do not generally allow computationally efficient algorithms for simple analysis. Thus, it leads to computationally intensive algorithms for image estimation/ classification problems. In addition, parameter identifications are not trivial tasks for MRF due to the computational complexity in the partition function [Besag, 1974; Potamianos, 1991]. Unlike conventional mono-scale MRFs whose per-pixel computational load typically grows with image size, multi-scale based image analysis has a per-pixel complexity independent of image size for optimal estimations. Thus, significant computational savings could be obtained on calculating optical flow [Chou, 1991] or the interpolation of sea level variations in the North Pacific Ocean from satellite measurements as in [Fieguth & Karl, 1995].

In Chapter 2, we explore a general approach for building multi-scale models on tree-structures. Figure 2-1 shows multi-scale tree structures for 1-D (dyadic) and 2-D cases (quadtree). For simplicity, our entire descriptions in Chapter 2 are carried out in the context of the dyadic tree which corresponds to the representation of l-D signals. The extension to higher dimensions (e.g. 2-D) is possible by trivial notational alternations without analytical complexities. For instance, in two-dimension (2-D) dyadic

tree would be replaced by a quadtree in which each node has four descendants instead of two, resulting in the same order of complexity per data point as in 1-D**.**

2.2 Multi-Scale Data Representation

2.2.1 State Space Models on q^{th} Order Tree

The models in [Chou, 1991; Luettgen, 1994] describe multi-scale stochastic processes indexed by nodes on a *tree*. (Figure 2-1) The key to our description is that multi-scale representations, regardless one-dimensional or higher dimensional signal, have a time-like variable, namely *scale*. Basically, all methods for representing and processing signals at multiple scales involve pyramidal data structures, where each level in the pyramid corresponds to a particular scale and each node at a given scale is connected both to a parent node at the next coarser scale and to several descendent nodes at the next finer scale. We commonly refer q^{th} order tree to a tree structure of nodes connected such that each node has q children (or offspring) nodes. Each node s is a scale index of state $x(s)$, where in general, the q^m state vectors at the mth level of the tree (for $m = 0,\ 1,\ 2, \cdots, M$) can be interpreted as representation of process at m^{th} scale. As described in [Chou, 1991], we define an upward (fine-to-coarse) shift operator B, where Bs is a *parent* of node s, and a set of downward (coarse-to-fine) shift operators α_i, $i = 1,\ 2, \ldots, q$ such that the q offspring of node s are $s\alpha_1, s\alpha_2, \cdots s\alpha_q$. Figure 2-2 illustrates the relations of $x(s)$, $x(Bs)$ and $x(s\alpha_1), x(s\alpha_2)$ in a second order tree.

Now, the dynamics of state $x(s)$ are then, modeled by the form of a Gaussian auto-regressive in scale

$$x(s) = \Phi(s)x(Bs) + \Gamma(s)w(s) \qquad w(s) \sim N(0, Q(s)) \tag{2.1}$$

This regression is initiated at the root node $s = 0$ with a state variable $x(0)$ having zero mean and covariance $P(0)$. The $w(s)$ represents white detail noise uncorrelated across scale and space, and also uncorrelated with the initial condition $x(0)$. This noise is assumed to be zero mean and covariance $Q(s)$. Since $x(0)$ and $w(s)$ are zero-mean, we can note that $x(s)$ is a zero-mean random process. Furthermore, since the detail noise $w(s)$ is white, the process $x(s)$ is characterized completely by $P(0)$ and the auto-regression parameters $\Phi(s)$ and $Q(s)$ for all nodes $s \neq 0$. The stochastic structure of multi-scale processes are expressed to provide an extremely efficient algorithm for estimating $x(s)$, based on noisy measurements $y(s)$. The corresponding measurement model is given by

$$y(s) = H(s)x(s) + v(s) \qquad v(s) \sim N(0, R(s)) \tag{2.2}$$

where $H(s)$ is the measurement mapping matrix and the noise $v(s)$ is white with covariance $R(s)$. It is uncorrelated with state $x(s)$ at all nodes on the tree.

Here, we can notice that the fine-to-coarse recursion corresponds to the multi-resolution *analysis* of signals whereas the coarse-to-fine recursion corresponds to the multi-resolution *synthesis* of signals because we add higher resolution details at each scale. Simply, (2.1) and (2.2) appropriately model the latter case for the multi-resolution signal processing.

2.2.2 Markov Property of Multi-Scale Process

It is well studied that the multi-scale stochastic model is analogous to traditional Markov time-series realization [Willsky, 2002]. The relation between two representations becomes clear once the Markov property of multi-scale processes is assumed [Bouman, 1994]. To describe the Markov property of multi-scale processes, we first assume that in

a qth order tree each node has a single parent node and q children nodes. This configuration separates the remaining nodes into $q+1$ subtrees (i.e. 3 subtrees for dyadic tree) by their children nodes and a parent node. Now, the Markov property states that if $x(s)$ is the value of the state at node s, then conditioned on $x(s)$ the states in the corresponding $q+1$ subtrees of nodes extending away from s are uncorrelated. For clarity, let Ψ_{si}, $i=1,2,3$ be the three subsets of states separated by s, a common node connected by three subsets of states, then Markov property states that

$$p_{\psi_{s1},\psi_{s2},\psi_{s3}\mid x_s}(\psi_{s1},\psi_{s2},\psi_{s3}\mid x_s)=p_{\psi_{s1}|x_s}(\psi_{s1}\mid x_s)\,p_{\psi_{s2}|x_s}(\psi_{s2}\mid x_s)\,p_{\psi_{s3}|x_s}(\psi_{s3}\mid x_s) \tag{2.3}$$

The property (2.3) implies that the tree processes are Markov in scale from coarse-to-fine. Also, the conditional pdf of the state at node s given the states at all previous scales depends only on the state at the parent node Bs. This is intuitive but important property in multi-scale signal processing, because it allows implementation of efficient inference (estimation) algorithm in tree structure. Specifically, it states that the sets of offspring nodes (e.g. two sets of nodes represented by yellow and green in Figure 2-3) extending from their common parent node (s) are decoupled by Markov property, thus the values of each sets of nodes can be estimated independently. This leads parallelized inference algorithm on tree (e.g. Multi-scale Kalman smoothing algorithm in Chapter 3). Figure 2-3 illustrates Markov property in dyadic tree.

The interpretation of multi-scale representation to the time-series realization is that the role of state information is to provide an information interface among subsets of the process [Willsky, 2002]. This interface must store just enough process information for conditional uncorrelation of corresponding process subsets. In the time-series case, this interface is between two subsets of the process (e.g., the past and the future), while in

the multi-scale case, the interface is among multiple (e.g. $q+1$) subsets of the process [Irving, 1995].

2.3 Conclusions

In Chapter 2, we have briefly reviewed a multi-scale signal model which leads to developing extremely efficient estimation algorithm. Due to the Markov property in (2.3) of tree structure, parallelized inference algorithm can be achievable.

The main importance of this study is to provide effective information fusion methods for disparately sensed stochastic process at different resolutions. The signal processing framework we summarized allows modeling of such multi-resolution data appropriately. This results in data fusion algorithms that are no more complex than algorithms for filtering single resolution data. As we discussed, the same can not be achieved by classical statistical signal model like MRFs [Bouman, 1994]. The use of a simple quadtree model has been widely exploited in image classification and reconstruction problems.

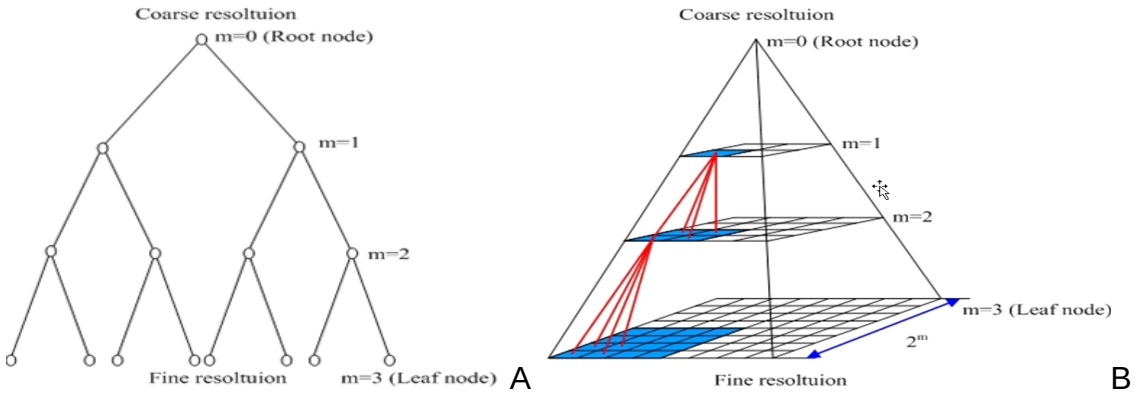


Figure 2-1. Multi-scale tree structures. A) 1-D dyadic structure, and B) 2-D quadtree structure. The root node corresponds to the coarsest scale while the leaf nodes comprise the finest scale.

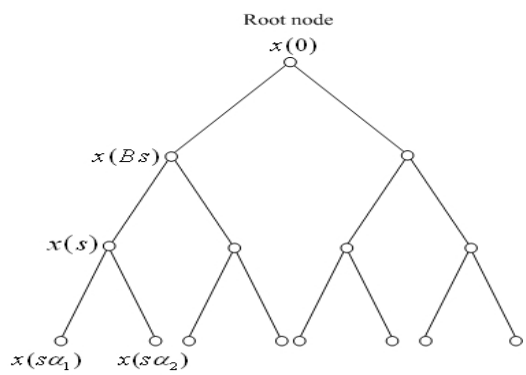


Figure 2-2. The first four levels of 2nd order tree structure (dyadic tree). The parent node of s is represented by Bs and two offspring are denoted by $s\alpha_1$ and $s\alpha_2$.

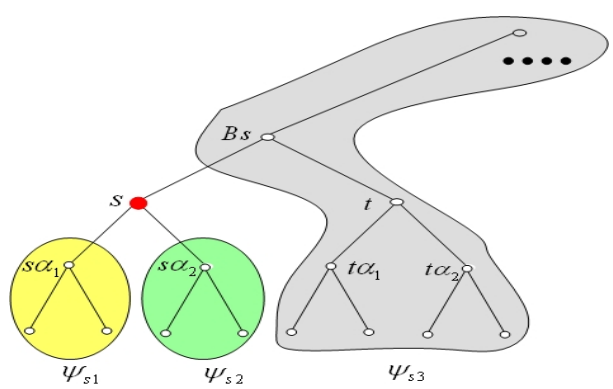


Figure 2-3. Markov property of dyadic tree structure. Conditioned on node s, the nodes in the corresponding 3 subtrees of nodes extending away from s are uncorrelated. (Each subset of nodes is represented by yellow for ψ_{s1}, green for ψ_{s2} and grey for ψ_{s3}, respectively)

CHAPTER 3
DATA FUSION BY MULTI-SCALE KALMAN SMOOTHING ALGORITHM

3.1 Introduction

In Chapter 2, we discussed multi-scale linear state model and performed some elementary statistical analyses on tree structure. Multi-scale model is driven by white noise, scale recursive and analogous to time domain Gaussian Markov-chain [Willsky, 2002]. In Chapter 3, we investigate the problem of optimal estimation involving this model, and develop the generalization of the Rauch-Tung-Striebel (RTS) algorithm [Brown, 1997]. Basically, RTS consists of two estimation steps; a fine-to-coarse filtering sweep followed by a coarse-to-fine smoothing sweep. The filtering step, corresponding to a generalization of the Kalman filter to multi-scale tree model, consists of three-step recursions; (1) measurement update, (2) fine-to-coarse prediction, and (3) the fusion of pixel information from fine-to-coarse scale. The step (3) has no counterpart in standard time-series Kalman filtering, and this leads to a new scale-recursive Riccati equation. [Chou & Nikoukhah, 1994]. This two-way sweeps estimation scheme optimally combines a stochastic multi-scale model with the available measurements at different resolutions according to a Kalman gain. Theoretical studies have proved that recursion based estimation method supports large classes of statistical processes in efficient way [Brown, 1997]. Like in time-series Kalman filter, the procedures of specifying prior information of a state space model must be preceded before implementing the multi-scale estimation algorithm.

In section 3.2, we present preliminaries for the development of MKS algorithm. In section 3.3 fundamental aspects of Kalman filtering/smoothing are discussed. The detail of MKS algorithm is described in section 3.4 and finally, section 3.5 gives conclusions.

3.2 Preliminaries

The Kalman filter is a best linear estimator when the states or variables to be estimated are Gaussian. There are several numbers of estimation methods, such as maximum likelihood (ML), maximum a posteriori (MAP), and minimum mean squared error (MMSE) [Kamen, 1999]. ML and MAP estimators both compute the conditional probability distribution functions (pdfs) and they try to find the most likely value of the random process given measurements. The MMSE estimator produces an estimate that is globally optimal since it seeks solutions by minimizing the variance of the ensemble of measurements [Kay, 1993]. The linear MMSE (LMMSE) estimate trades off between accuracy (global optimality) and computational complexity. Unlike the other methods enumerated here, LMMSE does not require computations of conditional densities, instead, it only depends on second order statistics of the process and measurements. Therefore, the LMMSE can be implemented as set of linear equations that can be solved with non-iterative fashion [Kay, 1993]. If the processes or the measurements are not normally distributed, i.e. they are not completely described by their first two moments (mean and variance), the LMMSE will represent a suboptimal estimator. However, even this is the case, the performance usually remains close to optimal (sub optimal) [Haykin, 2002]. In our fusion studies, the normalities (Gaussianity) of states and measurements are essentially assumed. For example, we assume that the topography observed by the Laser sensor (e.g. LiDAR) is Gaussian, hence, the estimates of elevation heights using these measurements are approximately Gaussian [Carter, 2001]. In [Johnson, 1998], a method is introduced to transform non-Gaussian distributed data to nearly Gaussian. Generally, the choice of this transformation depends on data type and would require additional processing steps.

For sections 3.3 ~3.4, we will derive more general problem of combining sparse and disparate data types at multiple scales. To achieve this goal, a Gaussian distributed state-space approach is chosen, leading a Kalman filter formulation [Grewal, 1993]. For better understanding of multi-scale Kalman filter, section 3.3 gives background on Kalman filter theory.

3.3 The Kalman Filter

The Kalman filter is a recursive stochastic estimator that attempts to minimize the mean squared error (MSE) between the estimates and the random variables being estimated [Kay, 1993]. Figure 3-1 depicts the recursive operations of the Kalman filter [Brown, 1997]. Since the solution is computed recursively using only estimate from the previous step and the present measurement, the standard Kalman filter has relatively low memory requirements. If the model input parameters are specified correctly, the Kalman filter is the optimal linear minimum mean-square error (LMMSE) estimator when the signal and noise distributions are Gaussian [Kay, 1993]. The Kalman filter conceptually makes balancing the uncertainty of the process with the uncertainty of the measurements in a linear combination to reach at estimate. As a result, it is able to handle sparse or missing data in optimal manner [Brown, 1997].

3.3.1 Linear State Model

The linear equations that relate a state and measurement in the Kalman filter are

$$\begin{aligned} x_{k+1} &= \Phi_k x_k + w_k \\ y_k &= H_k x_k + v_k \end{aligned} \tag{3.1}$$

where x_k and y_k represent a state and a measurement vector at step k, respectively. Φ_k is state transition matrix in state relation and w_k represents process noise. H_k is state-measurement mapping and v_k is measurement noise.

3.3.2 Kalman Filtering Algorithm

The recursion steps of Kalman filter are shown in Figure 3-1. In Figure 3-1, the process noise w is assumed to be white with covariance Q and the measurement noise v is also white and has covariance R. The process and measurement noises are not cross-correlated.

$$E[w_k w_i^T] = \begin{pmatrix} Q_k, \ i = k \\ 0, \ i \neq k \end{pmatrix}, \quad E[v_k v_i^T] = \begin{pmatrix} R_k, \ i = k \\ 0, \ i \neq k \end{pmatrix} \quad \text{and} \quad E[w_k v_i^T] = 0 \ \text{ for } \forall i, k \tag{3.2}$$

The conditions on w and v given by (3.2) must be necessarily specified for the Kalman filter to be operated as an optimal LMMSE estimator. It is also necessary that Φ, Q, H, and R in (3.1) and (3.2) are being specified. In practice, however, these parameters are not known exactly, but they are often known approximately. The performance of the filter depends on the qualities of parameter specification, however, in some cases, Kalman filter works well even when they are not perfectly known or the noise terms are not exactly white [Brown, 1997].

3.3.3 Kalman Smoothing Algorithm

The Kalman filter has been widely used to estimate the values of state variables given a model and their relationship to the measurements. It minimizes the trace of the error covariance matrix P in (3.3) which is equivalent to minimizing the mean squared error (MMSE) between the state x and the estimate $\hat{x}$, for each sample of independent variable [Brown, 1997].

In most of signal processing applications, the independent variable is temporally indexed or spatially coordinated. For the kth sample, the error covariance matrix is

$$P(k \mid k) = P_k = E[e_k e_k^T] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \tag{3.3}$$

where $E[\bullet]$ is the expectation operator. The kth estimate $\hat{x}_k$ is conditioned on the measurement at sample k (if present) and the previous estimate of the state $\hat{x}_{k-1}$. The Kalman equation naturally infers missing data because previously estimated state values are associated with measurements to determine the current state. The measurements and the state model are weighted according to their variances.

Kalman smoothing utilizes measurements to the current sample to improve the estimated state, in the sense of reduced variance [Brown, 1997]. The error covariance of the smoothed estimates is

$$\begin{aligned} P^s(k \mid N) &= P(k \mid k) + J(k)[P^s(k+1 \mid N) - P(k+1 \mid k)]J^T(k) \\ J(k) &= P(k \mid k)\Phi^T(k+1, k)P^{-1}(k+1 \mid k) \end{aligned} \tag{3.4}$$

where N is the total number of samples to be estimated, $k = N-1, N-2, \ldots, 0$ and Φ is the state transition matrix. The formulations in (3.4) are sometimes referred to as Rauch-Tung-Striebel (RTS) smoothing or fixed-interval smoothing [Grewal, 1993]. Basic recursion steps of RTS algorithm in time-series process are shown in Figure 3-2.

The derivation of multi-scale Kalman smoothing (MKS) is a generalization of classical RTS algorithm on tree structure introduced in Chapter 2, with the addition of a merge step to support the data at each scale. In addition to the estimates, it provides the corresponding uncertainty (error variance) for every estimate, which is useful for quantitative analysis.

3.4 Multi-Scale Kalman Smoothing (MKS) Algorithm

To derive the multi-scale Kalman filter, it is convenient to alter notation slightly. In 1-D process, k is used to denote the recursion index of the filter. Since multi-scale process is modeled on tree data structure, s will denote the node of the tree and replace k as the index of recursion. For a 2-D process, multi-scale Kalman smoothing begins at a fine-to-coarse sweep up to the quadtree which is analogous to time-series Kalman filtering. The only difference is merging step which propagates a posteriori information of finer scale to coarser scale. The filtering step is followed by a coarse-to-fine sweep down to the quadtree that corresponds to Kalman smoothing.

Using the scalar form, the course-to fine linear dynamic model is given by

$$\begin{aligned} x(s) &= \Phi(s)x(Bs) + \Gamma(s)w(s) \quad && \forall s \in S, s \neq 0 \\ y(s) &= H(s)x(s) + v(s) && \forall s \in T \subseteq S \end{aligned} \tag{3.5}$$

where x is the state variable, and y represents the measurements. The stochastic forcing function w is a Gaussian white noise process with unity variance, and the measurement error v is a Gaussian white noise process with scale dependent covariance matrix $R(s)$. S represents the set of all nodes on the quadtree, and T denotes nodes in which measurements are available. s is the node index on the tree, and $s = 0$ denotes the root node. B is a backshift operator in scale, such that Bs is one scale coarser than s. Φ is the coarse-to-fine state transition operator, Γ is the coarse-to-fine stochastic detail scaling function, H is the measurement-state mapping, and R represents the measurement variance of the measurements. Since a wide range of natural stochastic processes, such as topography, exhibits power law behavior in their power spectra, they can be effectively modeled as fractional Brownian motion (fBm) processes [Fieguth & Karl, 1995; Turcotte, 1997]. We therefore assume that our state

process (e.g. surface elevation) follows a $1/f^{\mu}$ model in scale [Slatton, 2001]. Using this model, the power spectrum of the state variable $x(s)$ is represented by the multi-scale model in (3.5) with specifying the coarse-to-fine state transition operator $\Phi(s)=1$ and process noise standard deviations in (3.6)

$$\Gamma = \Gamma_0 2^{(1-\mu)m/2} \tag{3.6}$$

The values of Γ_0 and μ are determined by first order regression matching of the power spectrum of measurement to a realization of the fBm model in log-log space. Since power spectra of discrete image data can only represent signal energy over a finite range of spatial frequencies, a 2-D Hamming window is applied to the data prior to computing the power spectra to reduce aliasing.

Let $w(s) \sim N(0,1)$, then $E[w(s)w(t)^T] = I\delta_{s,t}$ where $\delta_{s,t}=1$ for $s=t$. So, Γ^2 represents the variance of the stochastic detail that is incorporated as the resolution increases. The initial priors are specified for the process model that evolves downward. Assuming that a zero-mean process [Fieguth & Karl, 1995], the priors for the state mean and covariance are

$$\begin{aligned} \hat{x}_0 &= E\,[x(0)] = 0 \\ P_0 &= E\,[x(0)x^T(0)] = P_s(0) \end{aligned} \tag{3.7}$$

where $s=0$ represents a root node, and $P_s(s)$ denotes the covariance of the state at node s. These priors are used with the recursions in (3.5) to generate a realization of a multi-scale stochastic process. The resulting prior estimates of the state and state covariance at the leaf nodes are then used as the initial priors in the upward Kalman filter. In general, little is known about the process to be estimated, so $P_s(0)$ is chosen to

be some arbitrary large number [Gan, 2001]. The *a priori* process model in (3.5) is now completely specified. The corresponding upward model can be specified, which the Kalman filter will track

$$\begin{aligned} x(Bs) &= F(s)x(s) + \overline{w}(s) \\ y(s) &= H(s)x(s) + v(s) \\ F(s) &= P_s(Bs)\Phi^T(s)P_s^{-1}(s) \\ E[\overline{w}(s)\overline{w}^T(s)] &= P_s(Bs)[I - \Phi^T(s)P_s^{-1}(s)\Phi(s)P_s(Bs)] \\ &= Q(s) \end{aligned} \tag{3.8}$$

Here, Q is the process noise covariance in the upward model. It is the multi-scale analogous to Q in the time-series Kalman filter in section 3.3. F is the fine-to-coarse state transition operator.

The MKS algorithm proceeds with initialization, the upward sweep, and finally the downward sweep.

Initialization: At the leaf nodes, enter the prior values

$$\begin{aligned} \hat{x}(s \mid s+) &= 0 \\ P(s \mid s+) &= P_s \end{aligned} \tag{3.9}$$

Upward sweep: The upward sweep is equivalent to a Kalman filter operating in scale with an additional merge step. Using defined initial priors at the leaf nodes, the algorithm proceeds from the bottom of the quadtree up to the root node.

$$\begin{aligned} K(s) &= P(s \mid s+)H^T(s)[H(s)P(s \mid s+)H^T(s) + R(s)]^{-1} \\ \hat{x}(s \mid s) &= \hat{x}(s \mid s+) + K(s)[y(s) - H(s)\hat{x}(s \mid s+)] \\ P(s \mid s) &= [I - K(s)H(s)]P(s \mid s+) \end{aligned} \tag{3.10}$$

The projection step is applied at all scales from the bottom.

For quadtree, $i = 1,\ 2,\ 3,\ 4$

$$\hat{x}(s \mid s\alpha_i) = F(s\alpha_i)\hat{x}(s\alpha_i \mid s\alpha_i)$$
$$P(s \mid s\alpha_i) = F(s\alpha_i)P(s\alpha_i \mid s\alpha_i)F^T(s\alpha_i) + Q(s\alpha_i) \tag{3.11}$$

Each 2×2 group of nodes at the previous scale is merged into a single value at the current scale

$$\hat{x}(s \mid s+) = P(s \mid s+)\sum_{i=1}^{q} P^{-1}(s \mid s\alpha_i)\hat{x}(s \mid s\alpha_i)$$
$$P(s \mid s+) = [(1-q)P_s^{-1}(s) + \sum_{i=1}^{q} P^{-1}(s \mid s\alpha_i)]^{-1} \tag{3.12}$$

for the quadtree, $q = 4$.

Downward sweep: The upward estimate at the root node is used as the initial condition $\hat{x}^s(0) = \hat{x}(0 \mid 0)$ to start the downward sweep, where the superscript s refers to a *smoothed* quantity. The downward sweep then proceeds down the tree

$$\hat{x}^s(s) = \hat{x}(s \mid s) + J(s)[\hat{x}^s(Bs) - \hat{x}(Bs \mid s)]$$
$$P^s(s) = P(s \mid s) + J(s)[P^s(Bs) - P(Bs \mid s)]J^T(s) \tag{3.13}$$
$$J(s) = P(s \mid s)F^T(s)P^{-1}(Bs \mid s)$$

More detail descriptions of the MKS algorithm are found in [Chou & Nikoukhah, 1994; Fieguth & Irving, 1995]. This algorithm is non-iterative and has constant computational complexity per pixel with $O(s_M)$, where s_M is the number of nodes at the finest scale $m = M$. Figure 3-3 illustrates recursion steps of MKS algorithm in single quadtree. In MKS, filtering operation needs to associate all data at different scales and return to the finest scale to obtain a final result. Thus, the Kalman filtering is initiated from the finest scale and finished by Kalman smoothing at the finest scale.

Figure 3-4 shows an example of image fusion for a pair of multi-resolution image sets, a sparse TOPSAR (Topographic Synthetic Aperture Radar) at 11th scale and a dense ERS-1 (European remote sensing satellite) at 9th scale, using MKS algorithm.

The ERS-1 data set is served as a primary measurement because it covers wide areas without data void. At the finest scale, y in (3.5) and (3.10) will represent TOPSAR data, while at a coarser scale, it will represent ERS-1 data.

3.5 Conclusions

We have demonstrated MKS algorithm using dynamic models in (3.5). This framework leads to an efficient and highly parallelizable scale-recursive optimal estimation algorithm by generalizing the Rauch-Tung-Striebel smoothing algorithm on hierarchical structure. It involves the Kalman filter for the measurement update and fine-to-coarse prediction at each scale and followed by smoothing step for data filling process. An example has been illustrated to verify the potential of MKS algorithm to the multi-resolution data fusion and statistical regularization problem.

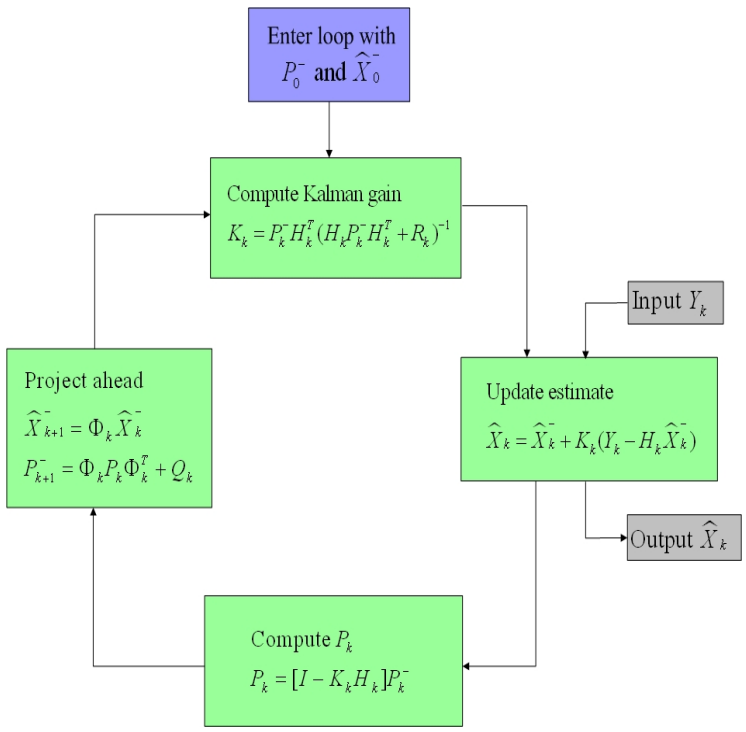


Figure 3-1. Recursion diagram of a Kalman filter

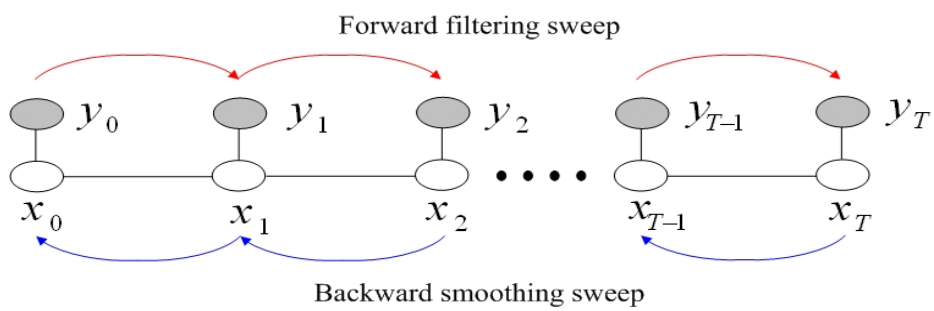


Figure 3-2. Two sweep steps of time-series Rauch-Tung-Striebel algorithm. 1) Forward sweep (Kalman filtering): Optimal inference on the hidden variables $x(t)$ given a collection of past and present measurements $y_0, y_{1,\ldots} y_t$, ,and 2) Backward sweep (Smoothing): Recursively computes the quantities of estimates given all measurements $y_0, y_{1,\ldots} y_T$.

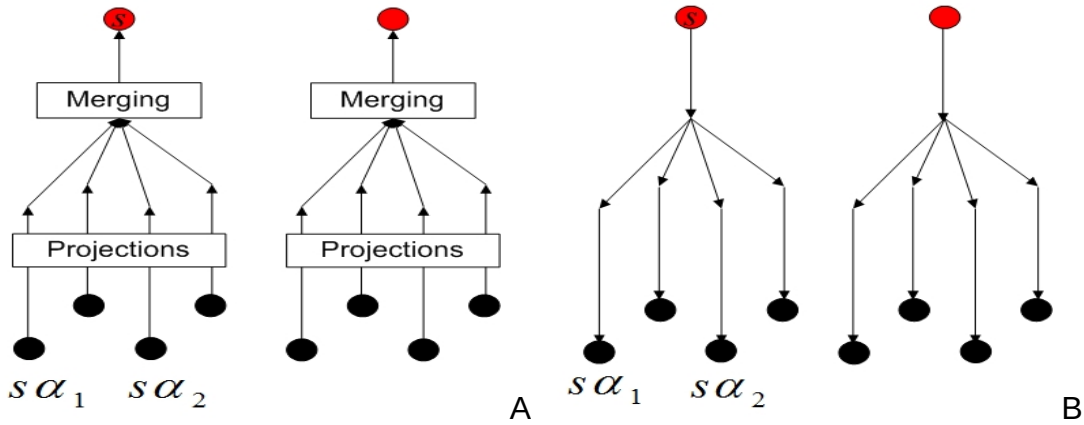


Figure 3-3. The MKS algorithm. A) Upward sweep (Kalman filtering) computes (3.9) ~ (3.12) from fine to coarse scale, and B) Downward sweep (Kalman smoothing) computes (3.13) from coarse to fine scale.

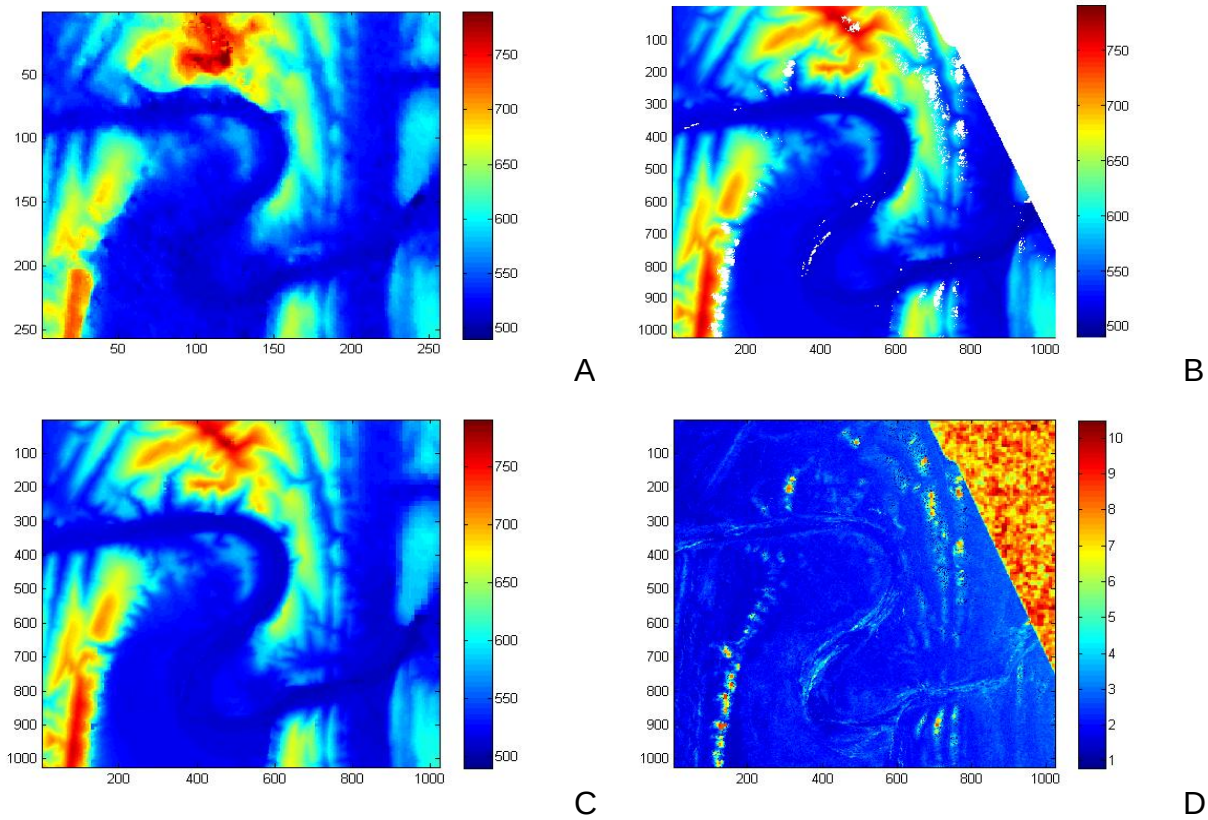


Figure 3-4. An example of image fusion by employing MKS algorithm. A) ERS-1 elevation data spaced by 20m, B) TOPSAR elevation data spaced by 5m, C) Fused estimate at the finest scale (11th scale), and D) RMS error of final estimate. All elevation units are in meters.

CHAPTER 4
APPLICATIONS

4.1 Introduction

We have known that the MKS algorithm is a globally optimal estimator for fusion of remotely sensed multi-resolution data in mean square error sense, and can be readily parallelized because of its Markov property. In some applications, however, we encounter the situations that the standard MKS algorithm in Chapter 3 is not applicable to solve image fusion problems. In such cases, we would consider new multi-resolution image fusion methods by exploiting properties in multi-scale framework. Thus, in Chapter 4, we will investigate more challenging image fusion cases by introducing new fusion algorithms which still yield efficiency and accuracy on estimation results.

The first application is that we are requested to fuse a large highly sparse fine-scale image using a dense coarse-scale image. In this study, we can obtain computationally moderate fusion scheme by exploiting Kalman and Markov properties of quadtree structure. This new implementation is referred to as Reduced-complexity MKS (RC-MKS) [Slatton, 2005].

The other fusion scenario arises when there are multiple numbers of measurements available at the same resolution (or scale). The adequate approach for this situation is to build a vector-valued MKS, but it would increase the dimension of the measurement model in (3.5). This results in accuracy problem due to an inversion of matrix whose dimension is prohibitively high. Therefore, we mitigate this computational difficulty by replacing covariance Kalman to information form for MKS algorithm which leads to iterative serial measurement update [Jhee, 2005]. The image fusion studies

explored in Chapter 4 are based on the works in [Slatton, 2005] and [Jhee, 2005], respectively.

4.2 Reduced-Complexity MKS (RC-MKS)

4.2.1 Motivations

Slatton, et al. [Slatton, 2000] successfully used MKS to fuse topographic elevations derived from high-resolution interferometric radar and airborne LiDAR data. However, in many remote sensing applications, it is desired to capture both regional and local structure. In such cases, the resolutions of the component data sets may differ by an order of magnitude or more, and the highest resolution measurements may be sparse relative to the coarse-scale measurements. Such is the case in coastal zone surficial mapping. For example, national data sets generally cover entire coastlines, while small state-funded acquisitions might cover only a few tens of square kilometers.

For a standard implementation of MKS in which two-dimensional surface elevation data are fused, the full set of recursive operations is performed at each node in the quadtree. This approach is highly inefficient if the finest scale (the set of leaf nodes in the quadtree) is sparsely populated with measurements. In a Kalman filter, the presence of a measurement (actually the measurement residual, known as the innovation) at a particular recursion index represents the injection of new information. If no measurement is present, the posterior estimate is simply the prior (predicted) estimate. In this work, multi-scale topographic and bathymetric data acquired over the Florida coast are fused together. We develop a reduced-complexity version of MKS to exploit this commonly faced sparse data configuration. We develop an efficient implementation of the MKS algorithm that indexes the leaf nodes so that only fine-scale nodes containing data are considered in the full recursion.

4.2.2 Sparse MKS Implementation

In data fusion applications where the measurement scales differ by an order of magnitude or more, it is quite common that the finest-scale data will only be available over a small subset of the quadtree leaf nodes. This case is shown in Figure 4-1. New information in presented to a Kalman estimator through the measurements, and in the absence of measurements, the prior estimates are simply propagated. Thus, we need only consider those subtrees that contain at least one leaf node at which a measurement is available (see Figure 4-2). Let $m = M$ be the level where dense coarse-scale measurements are available, and $m = M$ be the level where the finest-scale data are available. We define an indicator (flag) matrix to store the location of populated subtrees. The flag matrix is $2^N \times 2^N$, i.e. the size of the coarse-scale data matrix. To find valid subtrees, we tile the leaf nodes with a $2^{(M-N)} \times 2^{(M-N)}$ window and set the corresponding entry in the flag matrix to 0 if no node in the corresponding tile contains a measurement. Otherwise, the entry sets to 1. This can be accomplished at no extra computational cost during the specification of H. Using the flag matrix, we can obtain a range of row and column indices (R_M and C_M) at the finest scale for each valid subtree as

$$\begin{aligned} R_M &= \left[2^{(M-N)} R_N - \left(2^{(M-N)} - 1\right) : 2^{(M-N)} R_N \right] \\ C_M &= \left[2^{(M-N)} C_N - \left(2^{(M-N)} - 1\right) : 2^{(M-N)} C_N \right] \end{aligned} \tag{4.1}$$

where R_N and C_N are row and column indices of the flag matrix whose entries are set to 1. Due to the Markov property of MKS, each subtree can be processed independently. In practice, however, it is convenient to concatenate the subtrees, such that their bases

form a single rectangular matrix. That matrix supports $N_{st} \cdot 2^{M-N} \times 2^{M-N}$, where N_{st} is the number of subtrees. Other MKS parameters, such as H, Γ, R, are similarly permuted to correspond to the concatenated subtree structure. Subtrees that connect leaf node to scale N that do not contain measurements are not explicitly processed since the priors are propagated directly to scale N. The reduction in computational complexity afforded by this implicit pruning of the quadtree depends
on (1) the size of the subtree blocks, which is determined by the difference in scales M and N and (2) the aggregation of the fine-scale data, which determines the number of subtrees N_{st}. The reduction in floating point operations is determined by the reduction in the number of leaf nodes that must be processed in the recursion, which is given by

$$ME_{\%} = 100 - \left(\frac{N_{st} \cdot 2^{M-N} \cdot 2^{M-N}}{2^{M} \cdot 2^{M}} * 100 \right) \tag{4.2}$$

On the downward sweep, the full Kalman smoothing recursion need only proceed down the populated subtrees to the leaf nodes. The remaining coarse-scale estimates can be propagated to scale M via quadtree (nearest neighbor) or linear interpolation.

4.2.3 Image Fusion Results Using RC-MKS

This section presents the application of the proposed method for fusing data sets over the Florida coast. The topographic LiDAR [IHRC, 2004] measurements were available at a 5m grid spacing ($M = 13$), and the NGDC measurements [NGDC, 2005] were re-sampled from their original 90m spacing to 80m ($N = 9$) so the data sets would differ in resolution by integer powers of 4. Given these image pair, $M - N = 4$ and N_{st} is 48501, so the support of the subtrees at the finest scale consists of a 776016 × 16 array. From (4.2), we can calculate the expected computational savings. In the

standard MKS implementation, the number of leaf nodes to be processed is 2^{2M}, i.e. 8192^2 for $M = 13$. For the data sets in this work, the reduced leaf node set is $N_{st} \cdot 2^{M-N} \times 2^{M-N}$, i.e. $48501 \cdot 16^2$, which is a reduction in the number of nodes that must be processed of 81.5% at each scale below $m = N$.

It is also possible to measure the impact of pruning the quadtree in terms of memory required for a non-parallelized implementation of MKS. By reducing the number of subtrees between scale M and N that must be processed, we also significantly reduce the amount of memory required to store values of filter parameters and estimates at each level in the quadtree. Without loss of generality, we let a 1×1 matrix represent a single byte and compare how much memory is used for the MKS algorithm below scale N. The columns in Table 4-1 list the required memory for the up and down sweeps for standard MKS and the proposed method. The realized memory savings between the standard MKS that uses the full quadtree and the Reduced-Complexity MKS (RC-MKS) totals 81.4%, which corresponds well to the 81.5% savings predicted by (4.2). Using proposed method, we obtained a total reduction in floating point operations of 82.59% in the upsweep and 72.87% in the down sweep. The resulting fused surface elevations and its uncertainty (square root valued) are shown in Figure 4-3.

4.2.4 Conclusions

In this application, we investigated a large-scale image fusion problem whose computation loads are not afforded by standard MKS. When the data sets to be fused differ in resolution by an order of magnitude or more and the fine-scale measurements are heavily sparse, the standard MKS algorithm is inefficient. We reduce the number of

floating point operations per node and also reduce the number of nodes in the quadtree being processed. We have verified that this new implementation led to a dramatic reduction in computational complexity.

4.3 Vector-Valued MKS

4.3.1 Motivations

When multiple measurements are available at the same scale, the natural approach is to implement a vector-valued MKS. The major difficulty when fusing multiple data sets at particular scale is that the computational complexity grows as m^3 where m is the number of measurement sets. We ameliorate this problem by employing a serial measurement update in iterative way. We also perform the full MKS recursion only on nodes where measurements are available, thus reducing the number of a posteriori computation steps.

4.3.2 Standard MKS Using Vector-Valued Measurements

If we try to combine m measurements with a single MKS iteration using $m \times 1$ vector, speed and accuracy of the MKS algorithm suffer because large matrices must be processed, which increases computational load and round off error. For a scalar state variable $x(s)$, let $Y(s)$ be an $m \times 1$ measurement vector and $V(s)$ be its corresponding noise vector whose elements are independent. The measurement model in (3.5) then becomes

$$Y(s) = H(s)x(s) + V(s) \tag{4.3}$$

where $H(s) = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$ with dimension $m \times 1$. If we directly apply vector valued measurements to the Kalman filter recursions in [Chou & Benveniste, 1994], the Kalman gain expression is

$$K(s) = P(s \mid s+)H^T(s)(H(s)P(s \mid s+)H^T(s) + R(s))^{-1} \tag{4.4}$$

The expression in (4.4) requires an $m \times m$ matrix inversion, which becomes problematic for large m

4.3.3 The Information Form of Kalman Filter

Using the matrix inversion lemma [Strang, 1989], the standard (covariance form) Kalman filter equations can be permuted into an information form [Brown, 1997]. For a $m \times 1$ measurement vector, the error variance and Kalman gain can be written as

$$P^{-1}(s) = P^{-1}(s \mid s+) + \sum_{i=1}^{m} R_i^{-1}(s) \tag{4.5}$$

$$K(s) = P(s)[R_1^{-1}(s) \quad R_2^{-1}(s) \quad \cdots \quad R_m^{-1}(s)] \tag{4.6}$$

where $R_i(s)$ is i^{th} diagonal element of measurement noise covariance matrix. In (4.5) and (4.6), $H(s) = 1 \ \forall s \in T \subseteq S$ and is zero otherwise. The estimate is then

$$\hat{x}(s) = \hat{x}(s \mid s+) + K(s)(Y(s) - H(s)\hat{x}(s \mid s+)) \tag{4.7}$$

We note that two matrix inversions are required to obtain the inverse of the a priori error variance at each recursion step. The merging and projection steps remain the same as in the standard form of MKS.

4.3.4 Efficient Measurement Updates Methods

4.3.4.1 Serial measurement updates method

In case where $R(s)$, the covariance matrix of $V(s)$, is diagonal, it is possible to implement MKS using a serial measurement update method by treating each component of $Y(s)$ as a single independent measurement [Grewal, 1993]. By replacing

the standard Kalman equation in [Chou & Benveniste, 1994] with the information form [Brown, 1997], the main advantages are:

1) Reduced computation complexity

The number of arithmetic computations required to process an $m \times 1$ vector is dramatically reduced if we treat it as a collection of successive m scalar measurements.

2) Improved numerical accuracy

By modifying the MKS up-sweep algorithm, we can limit the opportunities for round off error caused by the matrix inversion in (4.4).

The filter implementation with this method requires m iterations of the measurement update using each row of $H(s)$ as a measurement mapping vector and the diagonal elements of $R(s)$ as the corresponding (scalar) measurement noise variances. The updating can be implemented iteratively in the following manner

For $i = 1, 2, \cdots, m$, define

$$P_{[i]}^{-1}(s) = P^{-1}(s \mid s+) + H_{[i]}^{T}(s) R_{[i]}^{-1}(s) H_{[i]}(s) \tag{4.8}$$

where $H_{[i]}(s)$ is the ith row of $H(s)$ and $R_{[i]}(s)$ is i^{th} diagonal element of $R(s)$.

Let $P_{[i]}(s)$ be the error variance associated with $H_{[i]}(s)$ and $R_{[i]}(s)$. Using (4.8), define the Kalman gain at i^{th} iteration as:

$$K_{[i]}(s) = P_{[i]}(s) H_{[i]}^{T}(s) R_{[i]}^{-1}(s) \tag{4.9}$$

Using expressions in (4.8) and (4.9), the sequentially generated Kalman gain after m iterations is

$$K(s) = [K_{[1]}(s) \quad K_{[2]}(s) \quad \cdots \quad K_{[m]}(s)]$$
$$= [C_{[1]}R_1^{-1} \quad C_{[2]}R_2^{-1} \quad \cdots \quad C_{[m]}R_m^{-1}] \tag{4.10}$$

where

$$C_{[i]} = (P^{-1}(s \mid s+) + R_{[i]}^{-1}(s))^{-1} \text{ for } i = 1, 2, \cdots, m \tag{4.11}$$

From (4.6) and (4.10), we note that the difference between the two expressions is the set of multiplication factors $C_{[1]}, C_{[2]}, \cdots C_{[m]}$ of each element in

$$[R_1^{-1}(s) \quad R_2^{-1}(s) \quad \cdots \quad R_m^{-1}(s)]$$

If we set every $C_{[i]}$ in (4.10) as

$$C_{[i]} = \left((P^{-1}(s \mid s+) + \sum_{i=1}^{m} R_i^{-1}(s) \right)^{-1} = P(s) \tag{4.12}$$

then the Kalman expressions in (4.6) and (4.10) become identical. So, at a scale where multiple measurements are available, (4.12) is computed once and substituted for $P_{[i]}(s)$ for all i in (4.9). Then iteration steps in (4.9) are repeated for $i = 1, 2, \cdots m$ to obtain each entry of the Kalman gain in (4.10) sequentially.

This approach is particularly useful when the number of recursions (nodes) is large, as it is for image fusion. For example, if k measurements are available at the jth scale in the quadtree, then the dimension of the measurement vector becomes $k \cdot 2^j \times 2^j$. As k and j increase, handling of measurement vector $Y(s)$ and other parameters in MKS algorithm becomes impractical by direct application of vector-valued MKS. Given prior estimate information $\hat{x}(s \mid s+)$, the innovation term in (4.7) is linearly combined with Kalman gain in (4.10) so that the estimate update equation can be rewritten as

$$\hat{x}(s) = \hat{x}(s \mid s+) + \sum_{i=1}^{m} K_i(s)(Y_i(s) - \hat{x}(s \mid s+)) \tag{4.13}$$

The summation term in (4.13) now can also be replaced with m iterations of update steps. For the ith iteration, the innovation associated with i^{th} element of the measurement vector is multiplied with gain $K_i(s)$ and stored for the next iteration update.

4.3.4.2 Selective measurement updates method

Typically, not every scale on the quadtree is populated with measurements. The error variance $P(s)$, Kalman gain $K(s)$ and update estimate $\hat{x}(s)$ at those scales do not require the full MKS recursion. In such a case, since $H(s) = 0$, we have

$$\begin{aligned} P^{-1}(s) &= P^{-1}(s \mid s+) \\ K(s) &= 0 \\ \hat{x}(s) &= \hat{x}(s \mid s+) \end{aligned} \tag{4.14}$$

We can simply merge and project information in (4.14) at the current scale upward so that it is used as prior information for the next coarser scale. This can significantly reduce the computation load when just a few scales are occupied with measurements. When the upward sweep reaches a scale where measurements are present, the normal recursion steps are resumed. All merged and updated estimates and error variances at each scale must be stored because these values are required for the downward smoothing process.

4.3.5 Image Fusion Results Using Vector-Valued MKS

Results are presented for fused data sets over the Finke River in Australia. The Finke River is located in Central Australia, where it serves as a long term study site for

fluvial geomorphology and paleo-hydrology. The river flows in a southerly direction between two mountain ranges and across lowland plains [Slatton, 2000].

Space-based InSAR data were acquired over the Finke River in 1996 from an ERS-1/2 tandem acquisition. The resulting DEM covers approximately 2,500 square kilometers and was processed to a 20*m* × 20*m* spatial posting. Multiple airborne InSAR images where later acquired in 2000 by the NASA/JPL TOPSAR sensor as part of the PacRim 2000 TOPSAR deployment. These flight-lines are approximately 10km × 50km with spatial resolutions of 5m. The TOPSAR lines were flown with different headings yielding diversity of viewing angle in regions of overlap [Slatton, 2001]. Relative to the ERS data, the TOPSAR data have higher spatial resolution and smaller height uncertainty; however, the large incidence angles, which are common in airborne SAR data, lead to numerous data dropouts due to radar shadowing, layover, and low-backscatter areas. We examine fusing ERS and TOPSAR DEMs to investigate the impact of resolution and multiple acquisitions on DEM quality in the context of different morphologic features such as elevation, gradient and curvature. The data sets consist of a single 512×512 ERS DEM and three 2048×2048 TOPSAR DEMs. (see Figure 4-4.) Coherence information was available so that measurement height uncertainty could be computed. Zebker, *et al*. [Zebker, 1992] previously derived the expression for standard deviation of height uncertainty (σ_h) from side looking geometric parameters and coherence information. Table 4-2 summarizes the system parameters and average standard deviations of error for each data set at finest scale. Each 2048×2048 TOPSAR image is treated as a single independent measurement and, the proposed MKS algorithm begins with following error variance equation at the finest scale;

$$P(S) = (P^{-1}(S \mid S+) + \sum_{i=1}^{3} R_{Top_i}^{-1}(s))^{-1} \tag{4.15}$$

where $R_{Top_i}^{-1}(s)$ is measurement noise variance of the ith TOPSAR DEM.

Also, once the updated estimate is obtained at the finest scale and projected to a coarser scale, the normal Kalman recursion steps in (4.5) ~ (4.7) are used in the remainder of the upward sweep. Standard deviations of the estimation error for MKS implemented with single TOPSAR images and with the vector-valued TOPSAR measurement vector are shown in Table 4-3. By employing serial and selective updating in sections 4.3.4.1and 4.3.4.2, we obtain significant computational complexity reduction. For quadtrees that contain data at just a two scales, this approach can lead to dramatic computational savings of 42% or more. The fused results at the finest scale (elevations) are shown in Figure 4-5.

4.3.6 Conclusions

We have examined a common data fusion problem, in which multiple sets of measurements are available at the same scale. We presented a method to reduce the operational complexities of MKS by introducing sequential and selective measurement updating. We show that DEMs resulting from the fusion of multiple data sets offer higher spatial resolution and lower height uncertainty. It shows the potentials to significantly improve the estimation of surface morphologic parameters, such as elevation, gradient, and curvature.

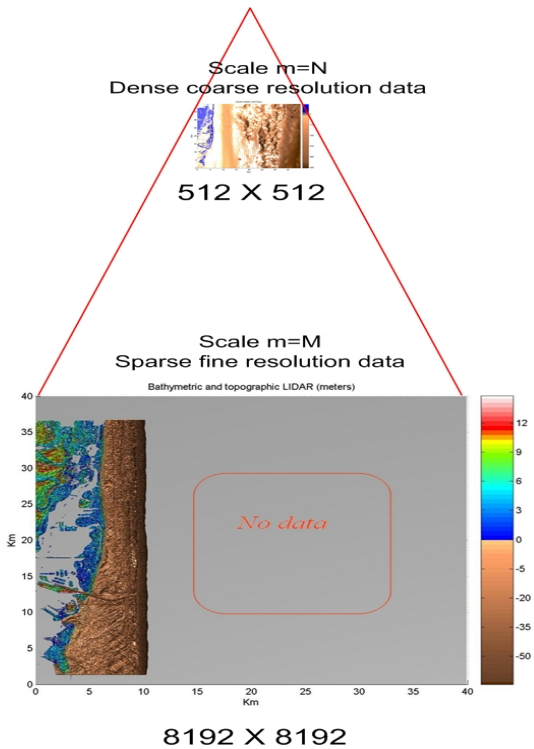


Figure 4-1. The finest resolution measurement is sparse relative to the coarse-scale measurement. The highly sparse fine-scale image is at 13th scale (5m spacing), and dense coarse-scale image is at 9th scale (80m spacing) on quadtree. The data voids are represented by grey. All elevation units are in meter.

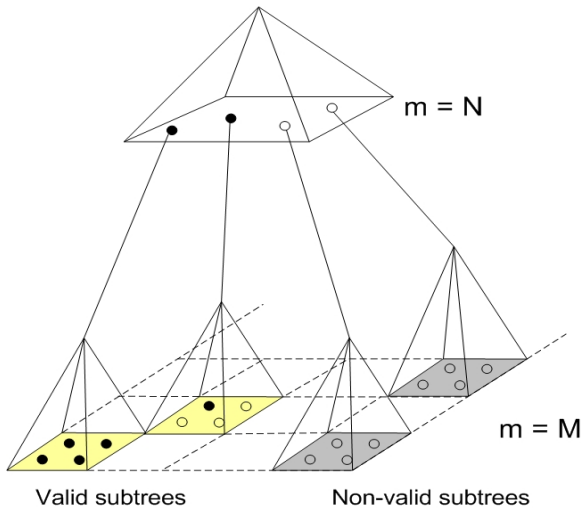


Figure 4-2. Set of subtrees for which finest-scale data are available. (each set of “valid” subtrees are represented by yellow)

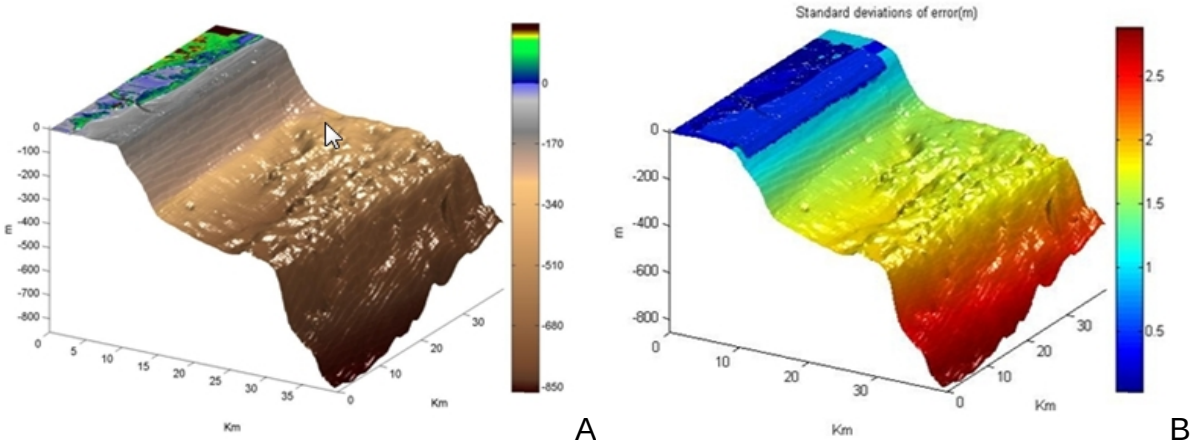


Figure 4-3. Perspective views of topographic and bathymetric elevations fused using the RC-MKS method. The coverage of area is 40 Km × 40 Km. A) Fused estimates (elevations) at 13th scale, and B) its RMS error. All elevation units are in meters.

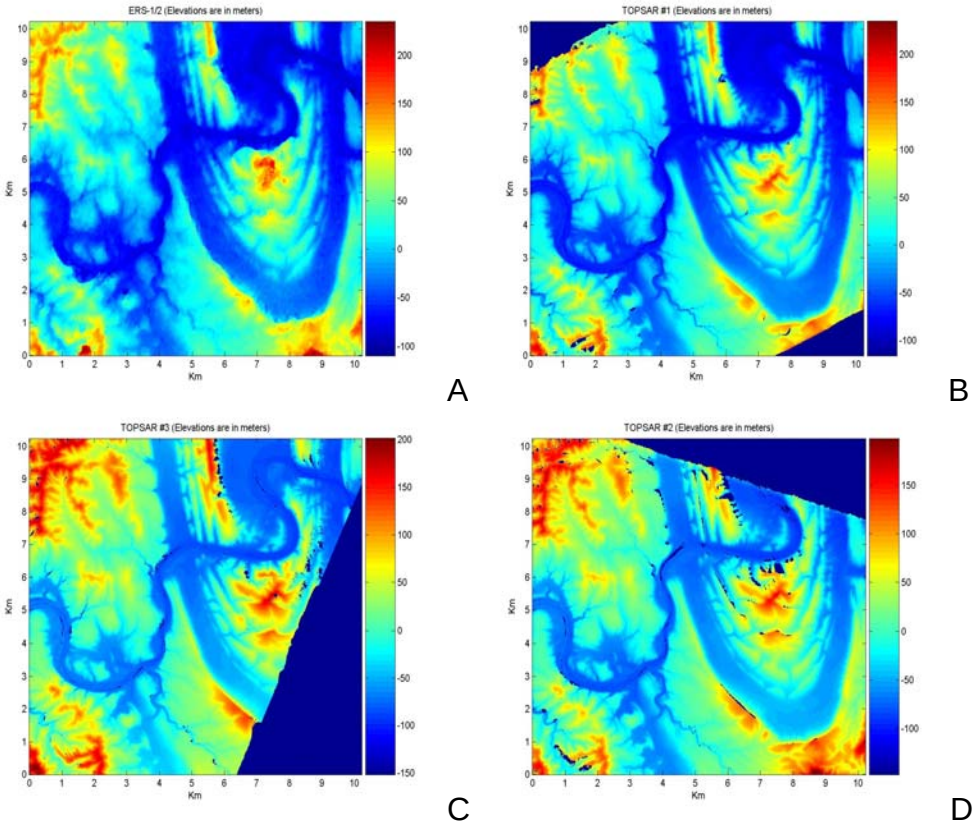
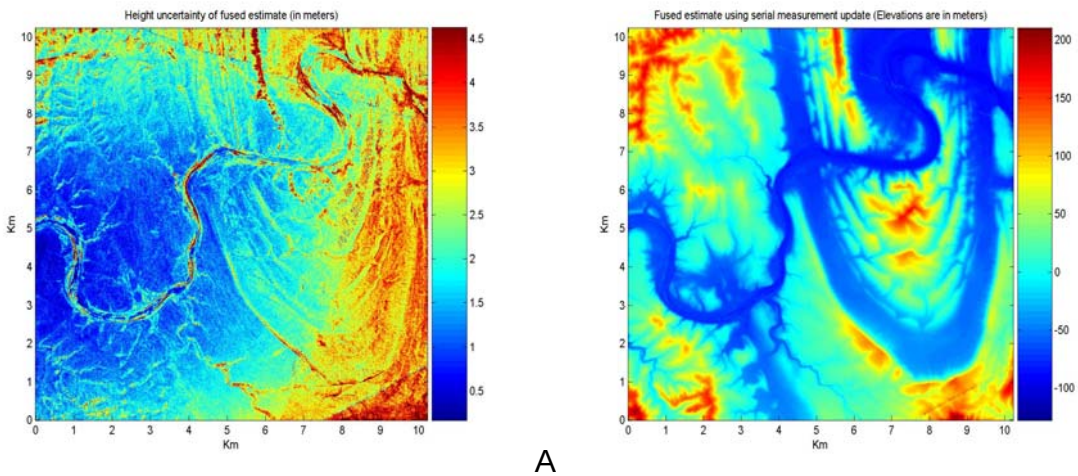


Figure 4-4. Image data sets to be fused. A) ERS-1 elevation data spaced by 20m, B) 1st TOPSAR elevation data spaced by 5m, C) 2nd TOPSAR elevation data spaced by 5m, and D) 3rd TOPSAR elevation data spaced by 5m. Dark blue area at each TOPSAR image represents data voids. All elevation units are in meters.



A B

Figure 4-5. Fusion results using vector-valued MKS. A) Fused estimate, and B) its height uncertainty using serial measurement update method. All elevation units are in meters.

Table 4-1. Comparison of memory storage using RC-MKS and standard MKS. (Mb:Mega bytes, Gb:Giga bytes)

m	Memory saving method		Standard method	
	Up	*Down*	*Up*	*Down*
10	25.22 Mb	13.19 Mb	136.31Mb	71.30 Mb
11	100.88 Mb	52.77 Mb	545.26 Mb	285.21 Mb
12	403.53 Mb	211.08 Mb	2.181 Gb	1.141 Gb
13	1.614 Gb	844.31 Mb	8.724 Gb	4.563 Gb
Total	3.27 Gb		17.58 Gb	

Table 4-2. ERS and TOPSAR system parameters and σ_h (RMS error variance)

Parameters	Data Types			
	ERS-1/2 (Spaceborne)	TOPSAR(Airborne)		
Base line (m)	136	5		
Number of multi-looks	5	9		
Operating wavelength (m)	0.0567	0.0567		
Base line angle (degree)	-6.7	65.08		
Flatform altitude (km)	793.6	7.73		
Slant range (near/far) (km)	851.2/861.4	13.61/23.84		
Incidence angle (degree)	18.37	25.36		
Averages of σ_h (m)	17.38	#1	#2	#3
		2.98	2.89	2.35

Table 4-3. Comparison of error performance using single TOPSAR MKS and the proposed method. (All elevation units are in meters.)

Data format used	Single TOPSAR data (#1,#2 and #3 in Figure 4-4)			Vector valued TOPSAR
	#1	#2	#3	
Average standard deviations of error (m) (RMS error variance)	1.79	2.01	2.12	1.22

CHAPTER 5
QUADTREE IMAGE MODEL AND BLOCKY ARTIFACT

5.1 Introduction

Multi-scale autoregressive model described in Chapter 2 provides an attractive alternative to traditional mono-scale Markov models. In such model, additional coarse-scale nodes are introduced to the fine-scale which may or may not be directly linked to any measurement. These nodes are hidden (or auxiliary) variables connected to describe the fine-scale stochastic process that is our primary interest. If we can properly design, the resulting tree structure can accurately model a wide range of stochastic processes. Despite of benefits enumerated in this dissertation, the most significant drawback revealed by tree structure is the presence of “blocky artifact” in the computed estimates (see Figure 5-1). The problem is caused by the fact that spatially adjacent, and hence supposedly highly correlated, fine-scale nodes may be widely separated in the quadtree structure [Fieguth & Irving, 1995]. As a result, dependencies (or correlations) between these nodes may be inadequately captured, causing blocky pixel artifact. One potential solution to remove blocky artifact problem is to build edges among all pairs of nodes at every scale. Such “in-scale” interactions should be able to account for short-range dependencies neglected by standard quadtree model.

5.2 Example: Comparisons of Covariance Structures among Three Different Graphical Models

5.2.1 Introduction

This example is exhibited to look at the source blockiness in our previous fusion studies (Figure 5-1). In this example, we compare the covariance structures of three different models: (1) mono-scale Markov chain, (2) dyadic multi-scale tree in Chapter 2, and (3) spatially interacted tree, namely pyramidal tree. Those three structures

(graphical models) are depicted in Figure 5-2. For Gaussian random vector $x = [x_1, x_2, \cdots, x_n]^T$, the joint pdf is parameterized by its mean (m_x) vector and covariance matrix (P). The joint pdf of Gaussian process x is then

$$p(x) \propto \exp(-\frac{1}{2} x^T \Theta x + z^T x) \qquad x \sim N^{-1}(z, \Theta) \tag{5.1}$$

$$\Theta = P^{-1} = (E[(x - m_x)(x - m_x)^T])^{-1}, \quad z = \Theta m_x \tag{5.2}$$

where joint pdf $p(x)$ is represented in information form and Θ is the information matrix which is equivalent to the inverse covariance matrix of random vector x (i.e. P^{-1}) [Kay, 1993]. If x is Markov with respect to graph, then the information matrix is sparse with the structure of specific graphical model. In Θ, the edge link between neighborhood nodes are represented by non-zero off diagonal. For smooth image field, we employ prior model (5.3) which controls the gradient quantities by adjusting the differences between the neighborhood nodes [Willsky, 2002]. If we denote the neighboring nodes of x_i as $N(x_i)$, then $p(x)$ in (5.1) becomes

$$p(x) \propto \exp(-\alpha \sum_{i \in V} \sum_{j \in N(x_i)} (x_i - x_j)^2) \quad \text{or} \quad p(x) \propto \exp(-x^T \Theta_{prior} x) \tag{5.3}$$

where V is a set of all nodes in graph structure. The parameter α controls how we can control the differences between a node and its neighborhood nodes. By (5.1) and (5.2), $z_{\mathrm{Prior}} = 0$. Suppose we have noisy measurement $y = Cx + v$, $v \sim N(0, R)$, then the conditional distribution x given y denoted by $p(x \mid y)$.

$$p(x \mid y) \propto \exp(-\frac{1}{2} x^T (\Theta_{\mathrm{Prior}} + C^T R^{-1} C) x + x^T (z_{\mathrm{Prior}} + C^T R^{-1} y)) \tag{5.4}$$

5.2.2 Estimation of Gaussian Process

The conditional distribution in (5.4) is also Gaussian such that $p(x \mid y) \sim N(\hat{x}, \hat{P})$. For Gaussian cases, given noisy measurement $y = Cx + v$, the conditional mean $\hat{x}$ is the best estimate of the unknown vector x under MMSE error criteria. Specifically, it is both the Bayes' least squares estimate minimizing the mean squared error ($\min_{\hat{x}} E\{(x-\hat{x})^2 \mid y\}$) and the maximum a posteriori (MAP) estimate maximizing conditional pdf in (5.4) [Kay, 1993]. The MAP estimate and its error variance are

$$\hat{x} = \arg\max(p(x \mid y)) = \Theta^{-1} h \tag{5.5}$$

$$\hat{P} = E\{(x-\hat{x})(x-\hat{x})^T \mid y\} = \Theta^{-1} \tag{5.6}$$

where $\Theta^{-1} = (\Theta_{prior} + C^T R^{-1} C)^{-1}$ and $z = z_{prior} + C^T R^{-1} y$. We can notice from (5.5) and (5.6) that if the number of variables (nodes on graph) becomes large, simple algebraic computations for optimal estimate $\hat{x}$ and its error variance $\hat{P}$ are not tractable since we must compute large size of matrix inversion (Θ^{-1}). For graph model without cycle (or loop) as shown in Figure 5-2 (B),however, there is an inference method which can compute optimal estimate and its error variance very efficiently. (e.g. MKS in Chapter 3).

5.2.3 Multi-Scale Modeling Using Pyramidal Tree

While tree structure introduced in Chapter 2 is more advantageous in capturing the correlation over mono-scale structure (or mono-scale MRF), its capability of capturing long range correlation is still limited by the lacks of inter connections between adjacent nodes in each scale. To overcome this limitation, pyramidal tree structure in Figure 5-2 (C) can be exploited. By allowing statistical links (statistical dependencies) between

pairs of neighboring nodes within scale in quadtree structure, the interactions among all nodes in graph would be captured well.

5.2.3.1 Prior model of pyramidal tree

In this section, we will describe the basic covariance structure of Gaussian pyramidal tree (Figure 5-2 (C)) using prior model introduced in (5.3). Using the obtained covariance structure, we will make comparisons of correlation decay with 1-D mono-scale (Figure 5-2 (A)) and dyadic tree (Figure 5-2 (B)). The result will show that pyramidal tree can capture longer correlation than both mono-scale and dyadic tree at finest scale, and does not produce blockiness experienced in dyadic tree model.

The prior model of a Gaussian graphical model can be represented by the corresponding information matrix $\Theta = P^{-1}$. The matrix Θ for the prior that we use in pyramidal tree structure consists of two components

$$\Theta_{\mathrm{Prior}} = \Theta_t + \Theta_s \tag{5.7}$$

where Θ_t accounts for the statistical relations between different scale, and Θ_s represents edges within scale. For pyramidal tree, Θ_t corresponds to a dyadic tree whose parent-child pair is connected by an edge, and Θ_s corresponds to nearest-neighbor grid models within scale (see Figure 5-3). We will describe the structures of Θ_t and Θ_s separately.

5.2.3.2 Intra-scale structure (For Θ_t)

In dyadic tree, we can consider a parent node and its two children nodes are connected by a statistical dependency. Let V_m be a set of all nodes in scale m where $m = 1,2,\cdots M$ and V_{m+1} is a set of nodes at one scale finer than m. Let $C(i) \subset V_{m+1}$ be a

set of children nodes whose parent node is $i \in V_m$ (nodes in a set of $C(i)$ in scale $m+1$ are just two children nodes of node i in scale m). Using prior model in (5.3), Θ_t is defined as follows:

$$\exp(-x^T \Theta_t x) = \exp(-\sum_{m=1}^{M} \beta_m \sum_{i \in V_m} \sum_{j \in C(i)} (x_{(m,i)} - x_{(m+1,j)})^2) \tag{5.8}$$

where the parameter β_m (Figure 5-4) determines how we control the difference between the value at a node at scale m and the values at each of its children node at scale $m+1$. Then Θ_t is decomposed by scale as follows:

$$\Theta_t = \begin{bmatrix} q\beta_1 I_{N_1} & \beta_1 \theta_{T_{12}} & 0 & 0 \\ \beta_1 \theta_{T_{21}} & (\beta_1 + c\beta_2) I_{N_2} & \beta_1 \theta_{T_{23}} & 0 \\ 0 & \ddots & \ddots & \ddots \\ 0 & 0 & \beta_{M-1} \theta_{T_{M,M-1}} & \beta_{M-1} I_{N_M} \end{bmatrix} \tag{5.9}$$

where N_m is the number of nodes at scale m, and I_{N_m} is the $N_m \times N_m$ identity matrix. The constant q is the order tree structure. $\theta_{T_{m,m+1}}$ is a sparse $N_m \times N_{m+1}$ matrix in which each entry corresponding to a parent-child pair. We denote the set of parameter β_m's as $\beta = [\beta_1, \beta_2, \cdots \beta_{M-1}]$.

5.2.3.3 Inter-scale structure (For Θ_s)

The nearest-neighbor grid model Θ_s is related with smoothness of image fields within scale. Since the statistical dependencies between different scales are modeled by Θ_t, every element of Θ_s that corresponds to scale by scale connection is zero. Inter-scale prior model θ_{sm} at scale m is

$$\exp(-x_m{}^T \Theta_{sm} x_m) = \exp(-\alpha_m \sum_{i \in V_m} \sum_{j \in N(i)} (x_{(m,i)} - x_{(m,j)})^2) \tag{5.10}$$

where V_m is a set of nodes in scale m and $N(i)$ is a set of neighborhood of node i in scale m. Then Θ_s is represented as

$$\Theta_s = \begin{bmatrix} \alpha_1\theta_{s1} & 0 & 0 & 0 \\ 0 & \alpha_2\theta_{s2} & 0 & 0 \\ 0 & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \alpha_M\theta_{sM} \end{bmatrix} \tag{5.11}$$

where dimension of each θ_{sm} for $m = 1,2,\cdots M$ is $2^{m-1} \times 2^{m-1}$. The diagonal elements of θ_{sm} are just equal to the number of neighbor nodes in scale m. The parameter α_m (Figure 5-4) determines the degree of smoothness of the field at scale m. If we want a smoother field, we can increase the value of α_m. We denote the collection of α_m's as $\alpha = [\alpha_1, \alpha_2, \cdots \alpha_M]$.

5.2.4 Comparisons of Correlation Decays (Mono-Scale, Dyadic Tree and Pyramidal Tree)

To compare the correlation characteristics of different graphical models, one-dimensional (1-D) process of length 64 is considered. The mono-scale graph is just 1st order Markov-chain model shown in Figure 5-5 (A), and we construct seven scales for both the dyadic (B) and pyramidal (C) trees. For the dyadic tree model, we used the same β_m ($m = 1, 2, \cdots 7$) parameters used in pyramidal tree, but remove all edges within scale (i.e. set $\alpha_m = 0$ for $m = 1, 2, \cdots 7$). For mono-scale Markov-chain, we use the parameter α_M of the pyramidal tree (M=7). Figure 5-6 represents the covariance matrices for each structure obtained by (5.6). Figure 5-7 shows the correlation decays derived by node distance between 8th node (i.e. $i = 8$) and node i, where i begins from

8 through 37 for the mono-scale Markov, dyadic, and pyramidal tree structures. Nodes distance is defined by the physical distance between 8th and i^{th} nodes. For example, 8th and 11th nodes have distance 3. Although dyadic structure (represented by blue curve) can capture longer correlation than mono-scale (represented by black curve), it shows "stair-like" shape. This turns out that the computed estimate using (5.5) in dyadic tree structure should have blocky embeddings where fine-scale data are not available. In this example, we have performed simple 1-D correlation analysis, but its extension to 2-D (i.e. quadtree) is straightforward. In addition, we have to note that this simulation results hold same in MKS algorithm since both MAP in (5.5) ~ (5.6) and Kalman filter/smoothing algorithm in (3.9) ~ (3.12) should have equivalent estimation results.

5.3 Conclusions

We have explored structural characteristics of three different types of graphical models namely mono-scale, quadtree and pyramidal tree. The illustrative example showed that quadtree structure can capture long range correlation among the nodes over mono-scale case, but its covariance structure exhibits blocky shape. Hence, MAP estimate of quadtree structure by (5.5) and (5.6) should suffer from blocky artifact. We also observed that efficient estimator can be implemented on quadtree structure with attractive computational cost (as shown in Chapter 3 and Chapter 4) but, blocky artifact shown on fine scale estimate is still problematic. Simply we can consider this result as a trade-off between algorithm efficiency and accuracy but in certain application, demanding for both visually and analytically improved result is more preceded. For Chapter 6, we will address this image blocky artifact problem in MKS algorithm using new scheme of fusion method.

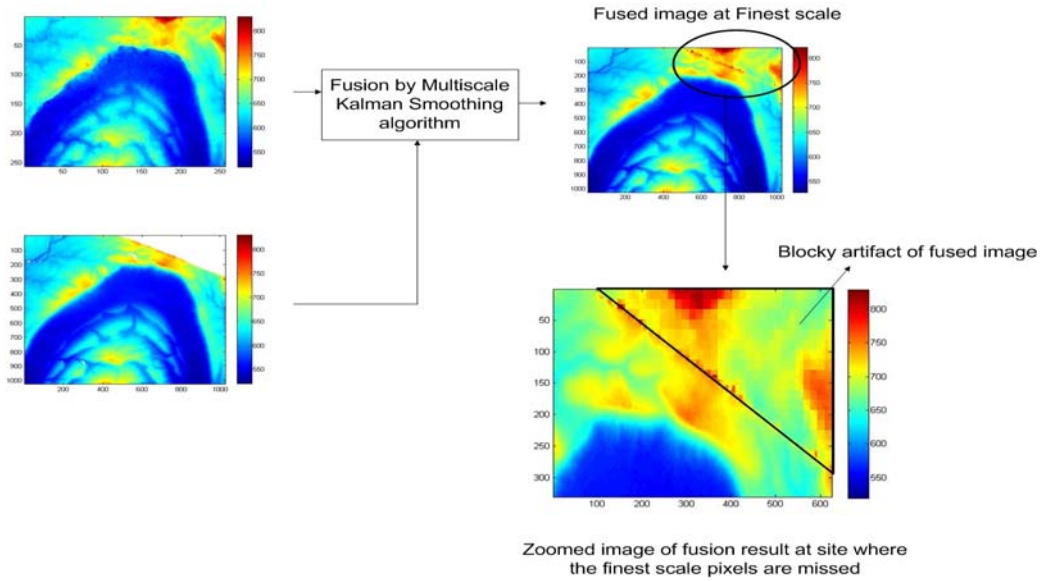


Figure 5-1.An example of blocky artifact. In this example, two images are being fused by MKS algorithm in Chapter 3. At final estimate, we can notice that there are blocky regions shown, and these appear at the pixel locations where fine-scale image pixel values are not available. (Data missing is represented by white at fine-scale image). All elevation units are in meter.

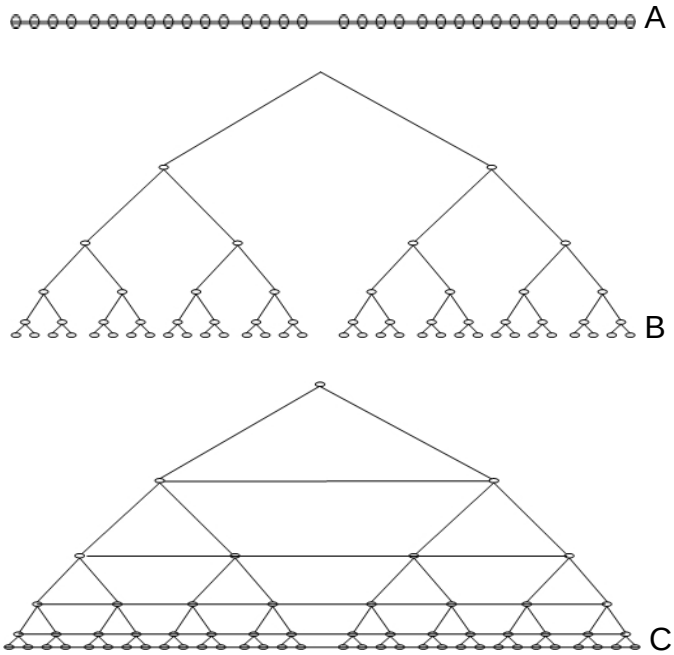


Figure 5-2. Three process models used in this example. A) mono-scale Markov chain, B) dyadic tree structure, and C) pyramidal tree structure. The edges (or connections) between nodes represent the statistical dependency between a node and its neighborhood ones.

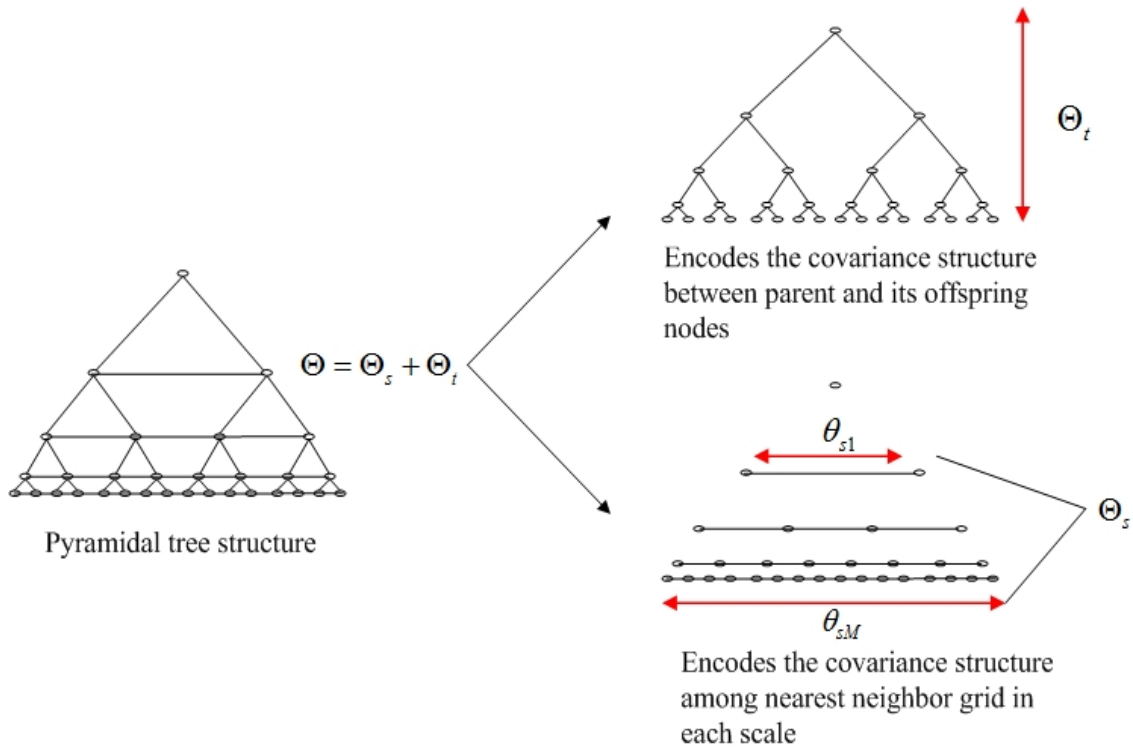


Figure 5-3. Inter-scale (Θ_s) and intra-scale (Θ_t) prior models in pyramidal tree structure.

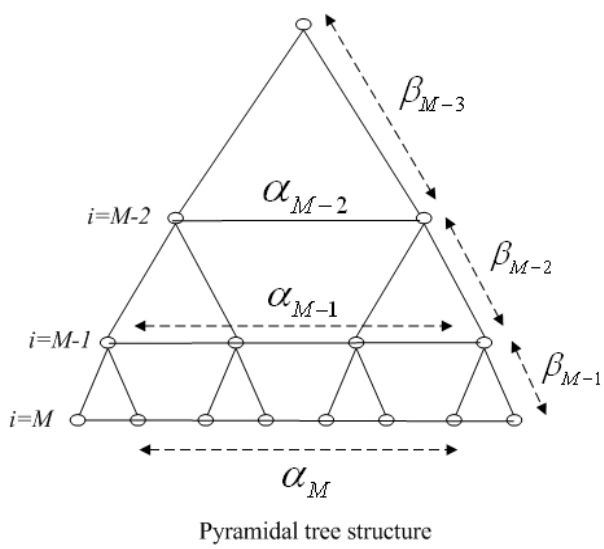


Figure 5-4. Pyramidal tree structure (1-D case). Penalizing parameters α_m and β_m of each scale are represented on pyramidal tree structure.

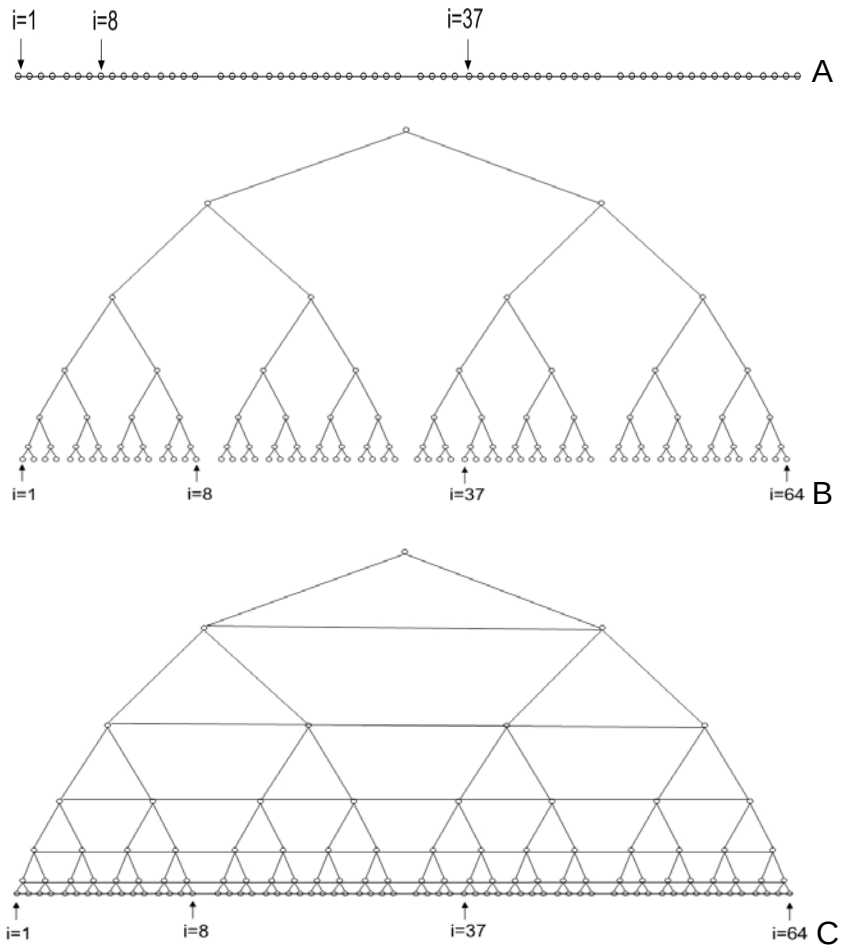


Figure 5-5. A) 1-D mono-scale Markov chain structure (64 variables), B) Dyadic tree structure (64 variables at finest scale), and C) pyramidal tree structure (64 variables at finest scale).

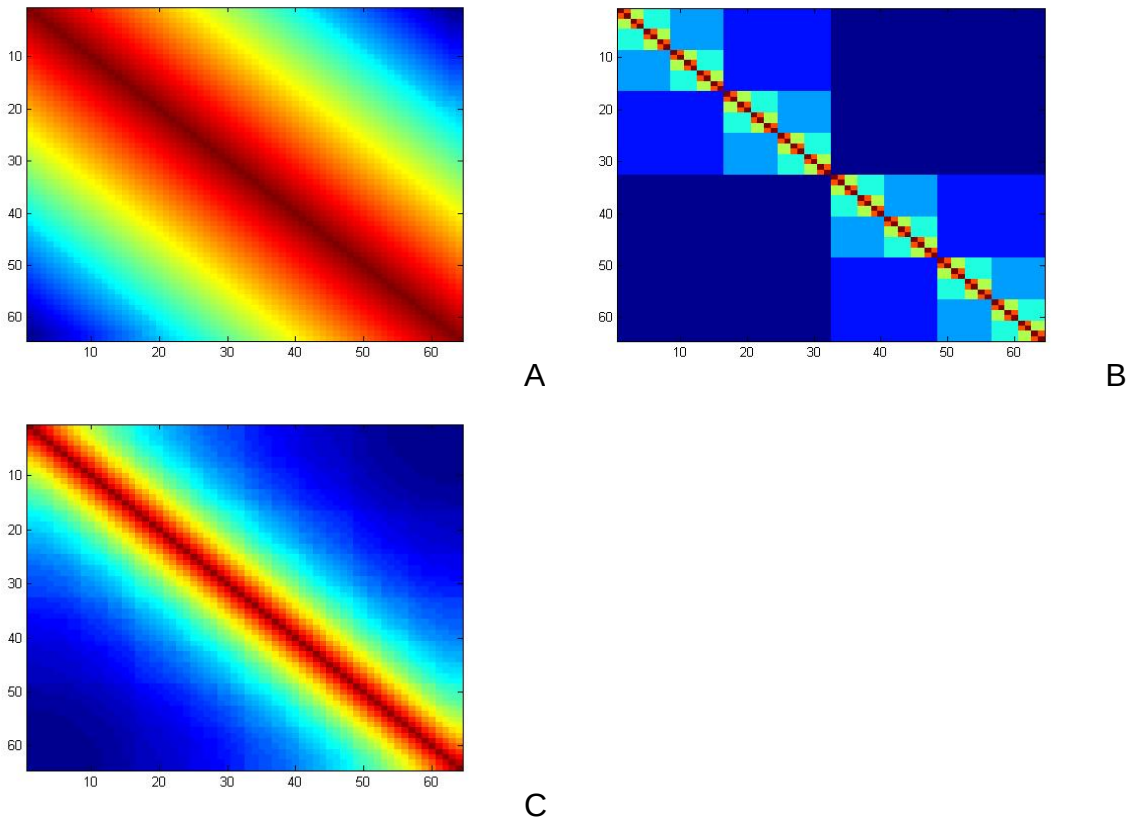


Figure 5-6. Obtained covariance matrices ($\Theta^{-1} = P$) in (5.6) using A) mono-scale Markov chain, B) dyadic tree, and C) Pyramidal tree.

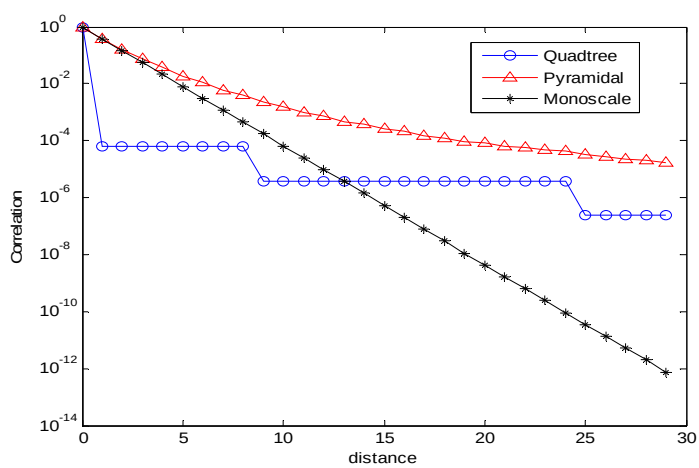


Figure 5-7. Correlation decay curves (all values are in log scale) of three different structures at the finest scale (1-D case). Parameters used in (1) Mono-scale: $\alpha_M = 1$, $\beta_m = \mathrm{N/A}$, (2) Dyadic tree: $\alpha_m = \mathrm{N/A}$, $\beta_m = [1\ 1\ 1\ 1\ 1\ 1\ 1]$, and (3) Pyramidal tree: $\alpha_m = [1\ 1\ 1\ 1\ 1\ 1\ 1]$ and $\beta_m = [1\ 1\ 1\ 1\ 1\ 1]$

CHAPTER 6
IMAGE FUSION USING SINGLE FRAME SUPER RESOLUTION

6.1 Introduction

Usually, MKS algorithm provides nice capability on estimating best pixel values when the fine-scale image pixels are available or size of missing area is small. However, if missing region of fine-scale image becomes large, the estimates at finer scale are simply driven by estimate values at coarser scale where dense low resolution image is populated (e.g. weighted by $J(s)$ in (3.13)). Since four children nodes at the finer scale are guided by corresponding parent node, the quantities of estimates at these offspring nodes become identical. As smoothing algorithm in (3.13) keeps continuing toward the finest scale, the interpolation (e.g. nearest) processes are repeated and these result in the "blocky-shapes" at final estimation image (at the finest scale). To describe this process, we will re-visit smoothing steps in MKS algorithm. In (3.13), scale recursive smoothed estimate and its error variance are computed by

$$\hat{x}^s(s) = \hat{x}(s \mid s) + J(s)[\hat{x}^s(Bs) - \hat{x}(Bs \mid s)] \tag{6.1}$$

In (6.1), it computes the smoothed estimate at scale s. At the regions where no data points are available, $\hat{x}(Bs \mid s) = 0$ since they are not supported by measurement updates during upward Kalman filtering in (3.10) ~ (3.11). That is, without measurement update, $\hat{x}^s(s)$ at that region is simply filled with the interpolated estimate values ($\hat{x}^s(Bs)$) at coarser scale. As shown in Chapter 5, the blockiness is caused by structural issues involved in the quadtree. In the view points of estimation algorithms, the MAP estimates using (5.5) and (5.6) are dominantly determined by model structure, hence the blocky covariance structure of quadtree originates the pixel blockiness. Since the MAP

estimator tends to output equivalent results to Kalman filtering and smoothing pair, same must be held in MKS algorithm.

Therefore, in Chapter 6 we will address blockiness issues by introducing more sophisticated fusion algorithm. The stem of this work is motivated by the fact that if we can provide the additional fine-scale image set which covers data void regions, then we would apply this newly generated image as an auxiliary measurement in fusion algorithm, thus solvable by vector-valued MKS introduced in section 4.2. Once this is properly performed, then in (6.1) $\hat{x}(Bs \mid s) \neq 0$ at data missing areas of the finer scale, thus detail information will be injected at fused estimates. This fusion scheme is theoretically sound because we will employ appropriate resolution enhancement technique which constructs most probable high resolution image within available training image sets and the resulting image will be applied to vector-valued MKS (in section 4.2) to provide additional high frequency information at fused result. Intuitively, if the quality of up-scaled (resolution enhanced) image is acceptable, it will nicely support measurement update in (3.10) at estimate where pixel values are not measured or degraded by noise or artifact.

To generate dense auxiliary high resolution image, image super resolution (SR) technique will be considered. The SR method is extensively used to reconstruct lost detail information during high to low resolution transformations. For example, it is useful in data de-compression applications since normal signal compression steps include the down-scale or re-sampling (down sampling) of original signal. When de-compression is performed, the lost high frequency components can be inferred by SR method.

In our fusion case, since we have only limited numbers of images, i.e. a pair of dense low and sparse high resolution images, learning-based SR technique will be under consideration. Learning process plays an important role in this implementation, because small local patch which has no counterpart high resolution patch will be learned by multiple sets of training image pairs (low-high training image sets). Then, the up-scaled patch is obtained by the linear combinations of corresponding high resolution patches whose low resolution counterparts are selected as similar patches with input patch. To find the set of similar patches, each input patch is represented in feature space and the K-Nearest Neighborhood (K-NN) searching algorithm [Duda, 2001] is exploited. It searches K most similar patches from training image sets by computing Euclidean distances between input feature vector and training feature vectors. Upon completing the computations of distances between input and all combinations of training feature vectors, only K nearest neighborhood patches are considered and their corresponding high resolution patches will be chosen to reconstruct the high resolution counterpart of input low resolution patch. To provide optimality in reconstructed high resolution patch, the weighting vector will be computed to score the similarities within K nearest patches. The computed weight vector is now linearly combined with high resolution patches found by K-NN method. These searching and reconstructing steps repeat for whole input image field and appropriate embedding constraints will be applied to satisfy the compatibility and smoothness between neighborhood patches in high resolution domain. Finally, the generated super resolved image is driven to the fusion algorithm introduced in section 4.2 to support the data void regions at fine-scale. We

refer this proposed fusion algorithm to as Super Resolution Multi-scale Kaman Smoothing algorithm (SR-MKS).

6.2 Super Resolution Method

Resolution enhancement methods based on simple smoothing and interpolation have been broadly used in image processing areas because of their computational conveniences. Smoothing is usually performed by utilizing various spatial kernel filters such as low pass, Gaussian kernel, Wiener, and median filters. Naïve interpolation methods, such as bicubic and spline interpolations, approximate an unknown point spread function (PSF) of a set of local patches to achieve higher resolution. Even though both methods are usually fast and easy to be implemented on practical applications, their basic operations are blending high frequency information in non-adaptive ways resulting in blurring effect in the output images. To overcome this limitation, edge sharpening techniques have been proposed to improve the results from interpolation methods [Greenspan, 2000; Morse, 2001]. These methods result in more realistic image by preserving the edges of objects in image fields, but may cause distracting highlighting artifacts. An alternate solution involving de-convolution technique by employing de-blurring filter [Mário, 2003; Shubin Zhao, 2003] provides quite good result, but it only enhances features that are present in the low resolution image. Recently, some image Super Resolution (SR) methods have been proposed by different literatures to provide realistic up-scaled version of images [Chang, 2004; Freeman, 2002; Freeman, 2000; Baker, 2000]. The basic idea of super resolution is to recover a high resolution image from single or multiple low resolution inputs. Broadly, it can be divided into two main categories as follows:

- The multiple-image Super Resolution: Each low resolution image introduces a set of linear constraints on the unknown high resolution counterpart.

- Example-Based Super Resolution: Correspondences between input low and unknown high resolution image patch pair are inferred from sets of low-high resolution training pair, and then recovers most likely high-resolution version of input image.

The concept of super-resolution was first introduced in [Tsai, 1984], using the frequency domain approach. A robust super resolution in [Kim et al.,1990] demonstrated the restoration of super-resolution images from noisy and blurred circumstance. More comprehensive approach to the super-resolution restoration problem was suggested by [Irani et al., 1991] and [Peleg, 1993], based on the iterative back projection method. [Joshi and Chaudhuri, 2003] proposed a learning-based method for image super-resolution from zoomed measurements. They define the high resolution image as a Markov random field (MRF) and, the model parameters are trained from the most enlarged measurement. [Bishop et al., 2003] proposed more sophisticated learning based super-resolution enhancement for video clip processing.

6.3 Super Resolution Multi-Scale Kaman Smoothing Algorithm (SR-MKS)

The resolution enhancement method proposed in this section is classified under learning-based super-resolution. Let a low-resolution image be an input to image super resolution system. Then, its high resolution pair will be reconstructed by employing multiple numbers of low-high resolution training image pairs. Specifically, if we are able to search a set of low resolution patches from training sets whose local geometries are similar to an input patch, their corresponding high resolution patches will also have similar local geometries with up-scaled version of input low resolution patch. To satisfy

this requirement, we will define local geometric features and search the most similar patches within training sets.

6.3.1 Summary of Super Resolution Algorithm

The learning-based image SR algorithm is summarized as follows. Given an input low-resolution image L_s, we estimate its counterpart high-resolution image $\hat{H}_s$ by multiple training low-high resolution image pairs L_t's and H_t's. We assume that low-high resolution image pair consists of a set of small overlapping image patches and the number of small patches consisting of L_s and $\hat{H}_s$ pair is same. This holds for each of L_t and H_t pair as well. For notational convenience, we define the sets of small image patches as $\mathbf{L_s}$, $\hat{\mathbf{H}}_\mathbf{s}$, $\mathbf{L_t}$ and $\mathbf{H_t}$, respectively.

$$\begin{aligned}
\mathbf{L_s} &= \{ l_s^1, l_s^2, \cdots, l_s^{N_s} \} \\
\hat{\mathbf{H}}_\mathbf{s} &= \{ \hat{h}_s^1, \hat{h}_s^2, \cdots, \hat{h}_s^{N_s} \} \\
\mathbf{L_t} &= \{ l_t^1, l_t^2, \cdots, l_t^{N_t} \} \\
\mathbf{H_t} &= \{ h_t^1, h_t^2, \cdots, h_t^{N_t} \}
\end{aligned} \tag{6.2}$$

where l_s^n, $\hat{h}_s^n$, l_t^m and h_t^m are small patches embedded in L_s, $\hat{H}_s$, L_t and H_t. N_s is number of patches defined in input low-high resolution images and N_t is total number of patches in training image pairs. N_s, N_t and the degree of patch overlapping depend on the implementation. We assume that each estimate patch $\hat{h}_s^n$ needs to not only be related with corresponding patch l_s^n, but it should also appropriately preserve some inter-patch relationships with neighborhood patches $\hat{h}_s^{n-1}$ and $\hat{h}_s^{n+1}$. The former satisfies the accuracy of patch similarity matching and the latter defines the local smoothness constraints of the estimated high-resolution image. By satisfying these requirements as

close as possible, we can claim that any patch $\hat{h}_s^n$ reconstructed by patches in $\hat{\mathbf{H}}_\mathbf{t}$ holds similarity matching with multiple patches learned from the training set and local relationship between patches in $\mathbf{L}_s$ should be preserved in $\hat{\mathbf{H}}_\mathbf{t}$. Furthermore, neighboring patches in $\hat{\mathbf{H}}_\mathbf{t}$ should have smoothness constraint through overlapping compatibility.

Let $\mathbf{F_{Ls}}$, $\mathbf{F_{\hat{H}s}}$, $\mathbf{F_{Lt}}$ and $\mathbf{F_{Ht}}$ be the sets of defined feature vectors for patches in $\mathbf{L_s}$, $\hat{\mathbf{H}}_\mathbf{s}$, $\mathbf{L_t}$ and $\mathbf{H_t}$, respectively.

$$\begin{aligned}
\mathbf{F_{Ls}} &= \{ f_{Ls}^1, f_{Ls}^2, \cdots, f_{Ls}^{N_s} \} \\
\mathbf{F_{\hat{H}s}} &= \{ f_{\hat{H}s}^1, f_{\hat{H}s}^2, \cdots, f_{\hat{H}s}^{N_s} \} \\
\mathbf{F_{Lt}} &= \{ f_{Lt}^1, f_{Lt}^2, \cdots, f_{Lt}^{N_t} \} \\
\mathbf{F_{Ht}} &= \{ f_{Ht}^1, f_{Ht}^2, \cdots, f_{Ht}^{N_t} \}
\end{aligned} \tag{6.3}$$

If l_s^n and $\hat{h}_s^n$ have similar local geometries, their corresponding feature vectors f_{Ls}^n and $f_{\hat{H}s}^n$ are closely located in feature space. More specifically, if we can extract appropriate l_t^m from $\mathbf{L}_t$ whose feature vector f_{Lt}^m is close enough to f_{Ls}^n, then the distance between f_{Ht}^m and $f_{\hat{H}s}^n$ are also close in feature space. For each f_{Ls}^n of l_s^n, the algorithm is then generalized by how we can find closest f_{Lt}^m's, and finally it follows finding its high resolution feature pairs f_{Ht}^m and $f_{\hat{H}s}^n$ whose corresponding patches are in $\mathbf{H_t}$ and $\hat{\mathbf{H}}_\mathbf{s}$, respectively. To search candidate similar patches of input patch l_s^n, we employ K-Nearest Neighborhood (K-NN) algorithm which searches most K closest feature vectors $\mathbf{K} = \{\tilde{l}_t^1, \tilde{l}_t^2, \cdots \tilde{l}_t^K\}$ by minimizing the l_2 norm between f_{Ls}^n and all available feature vectors in $\mathbf{F_{Lt}}$. Once the computations of l_2 norm are complete in feature space, it seeks the K corresponding patches $\tilde{l}_t^1, \tilde{l}_t^2, \cdots \tilde{l}_t^K$ from training image sets whose l_2 norm

are K smallest. The example of K-NN algorithm is illustrated in Figure 6-1 and Figure 6-2. In this example, the size of input patch is 5×5. Figure 6-1 illustrates the K (K=4) most similar patches found within 8 training low-high image pairs (Figure 6-3). We consider the 4 times magnification of original low resolution patch, thus it is equivalent to obtain patch up-scaled by 2 in quadtree. Figure 6-2 shows the corresponding estimated high resolution counterpart of input patch in Figure 6-1. As we expected, the searching algorithm employed successfully finds the up-scaled version of original input patch. Based on the patches searched by K-NN algorithm, we need to compute the best reconstruction weight vector $\Gamma_n = \{\gamma_n^1, \gamma_n^2, \cdots \gamma_n^K\}^T$ to construct the super resolved patch. To hold optimality, the weight vector $\Gamma_n = \{\gamma_n^1, \gamma_n^2, \cdots \gamma_n^K\}^T$ is obtained by minimizing the local error ε_n for each patch l_s^n.

$$\min_{\Gamma_n}(\varepsilon_n) = \min_{\Gamma_n}\left\| l_s^n - \sum_{q=1}^{K}\gamma_q \tilde{l}_t^q \right\|^2 \quad \text{where } \Gamma_n = \{\gamma_n^1, \gamma_n^2, \cdots \gamma_n^K\}^T \quad \text{s.t} \quad \Gamma_n^T\Gamma_n = 1 \tag{6.4}$$

By solving minimization problem, we can obtain

$$\Pi_n = (l_s^n 1^T - \mathbf{K})^T (l_s^n 1^T - \mathbf{K}) \quad \text{and} \quad \Gamma_n = \frac{\Pi_n^{-1}1}{1^T \Pi_n^{-1} 1} \tag{6.5}$$

where l_s^n and $\tilde{l}_t^q$ have dimensions of $q^2 \times 1$ (q is size of patch) and now they are represented as vectorized versions of l_s^n and $\tilde{l}_t^q$, respectively. 1 is a $K\times1$ unit column vector. Then, the corresponding up-scaled patch estimate $\hat{h}_s^n$ is obtained by

$$\hat{h}_s^n = \sum_{q=1}^{K}\gamma_n^q \tilde{h}_t^q \tag{6.6}$$

where each element in $\hat{\mathbf{K}} = \{\tilde{h}_t^1, \tilde{h}_t^2 \cdots \tilde{h}_t^K\}$ is corresponding high resolution patches in $\mathbf{K} = \{\tilde{l}_t^1, \tilde{l}_t^2, \cdots \tilde{l}_t^K\}$. Finally, we enforce local compatibility and smoothness constraints between adjacent patches $\hat{h}_s^{n-1}$ and $\hat{h}_s^{n+1}$ in the estimated high-resolution image through overlapping.

6.3.2 Feature Representation

As discussed above, each image patch is represented by feature vector. In this section, we will consider about the feature representations for both low and high-resolution images. For the low-resolution images, we use the relative intensity changes of pixels in patch because it captures more detail configurations of shape in the patch than simple intensity (or elevation) values. We choose the first-order and second-order gradients of each pixel in patch to represent. For example, in Figure 6-4, first and second order gradients vector of x_{33}, can be easily be derived as follows:

1st gradient at x_{33}: $\begin{bmatrix} x_{34} - x_{32} \\ x_{43} - x_{23} \end{bmatrix}$ (6.7)

2nd gradient at x_{33}: $\begin{bmatrix} x_{35} - 2x_{33} + x_{31} \\ x_{53} - 2x_{33} + x_{13} \end{bmatrix}$ (6.8)

By concatenating two computed gradient vectors together, we obtain four feature values for each pixel. For instance, if $p \times p$ patch is used, the dimension of corresponding feature vector becomes $4p^2$. For high resolution patches, we define the features for each patch based only on the intensity (elevation) values of the pixels. Since the features used for the low-resolution patches do not reflect the elevation values, we subtract the mean value of each trained low resolution patch $\tilde{l}_t^m$ from

corresponding high resolution patch. When we obtain $\tilde{h}_t^n$, the mean elevation value of input low patch l_s^n will be added.

6.3.3 Super Resolution Multi-Scale Kalman Smoothing (SR-MKS)

Using the super-resolution techniques introduced in sections 6.3.1 and 6.3.2, an auxiliary high resolution image will be generated. For accurate result, we choose the smoothed estimate $\hat{x}^s(s_{coarse})$ at coarse scale, i.e. scale where dense low resolution image is populated, as an input image for super resolution process. Thus, we run standard MKS algorithm to obtain the smoothed estimate at coarse-scale prior to applying super resolution algorithm. This is reasonable approach because the smoothed estimate at coarse-scale already contains blended information of available images within tree structure. Alternatively, we can consider the up-scaling of un-fused low resolution image (e.g. noisy and no fine-scale information is fused), but poor sensor often provides incorrect information of target of interest. If this is the case, even if we can successfully enhance the resolution of this image, the obtained result will have chance to produce completely erroneous version of high resolution image. Furthermore, fused estimate at data missed area is directly derived by estimated values from coarse scale. So, if successful resolution enhancement is made on the smoothed estimate at coarse-scale, it will guarantee that the resulting image holds optimality carried by MKS fusion algorithm.

6.4 Simulations

To verify the performance of proposed method, real terrain elevation data sets are used. In this simulation, we use 5 $\times$ 5 local patch with an overlap of four pixels between adjacent patches, hence the number of overlapped pixels among the high resolution

patches is16. Also, we designed K-NN with K=4. The low-high image pair used in this section is cropped different site from image sets used in section 4.2. Figure 6-5 shows the input image data sets for SR-MKS algorithm. The low resolution image at 9th scale (256 $\times$256 and 20m spacing) and incomplete high resolution image at 11th scale (1024 $\times$1024 and 5m spacing) are being fused by standard (as in Chapter 3) and proposed methods. To construct vector-valued MKS, the smoothed estimate at 9th scale is computed using standard MKS and super-resolved by algorithm in 6.3.2. For each high resolution training image in Figure 6-3, we obtain the corresponding low resolution training image by down sampling order of 4. Figure 6-6 compares the fusion results using standard MKS algorithm and SR-MKS. To illustrate details of fused results, we show estimates at some zoomed areas where fine-scale image is void. Since the difference between coarse and fine scales is 2 (4 (2^2) in resolution), 4 times pixel magnification is performed based on the algorithm in 6.3.2 and 6.3.3. Table 6-1 summarizes the MSE performances of fusion results. The simulation results show that SR-MKS successfully works on suppressing blocky artifact. The MSE performance obtained by new method is better than standard MKS. This new fusion method by exploiting image super resolution technique shows very pleasing empirical results in the senses of analytical and visual improvements.

6.5 Conclusions

We have proposed a new multi-resolution fusion scheme by employing the concept of single-image super resolution technique. The resolution enhanced image is applied to the fusion algorithm as a new source of measurement to mitigate the blockiness. More specifically, gaining of resolution at data missing area of fused image

does not depend only on a pair of input low-high resolution images. Instead, it is brought simultaneously by multiple high input resolution images, i.e. original incomplete and generated (super resolved) high resolution image pair. We verified the performance of the proposed scheme by real data simulation. It shows that the proposed SR-MKS provides better estimation results in the senses of both MSE performance and visual improvement over standard MKS algorithm.

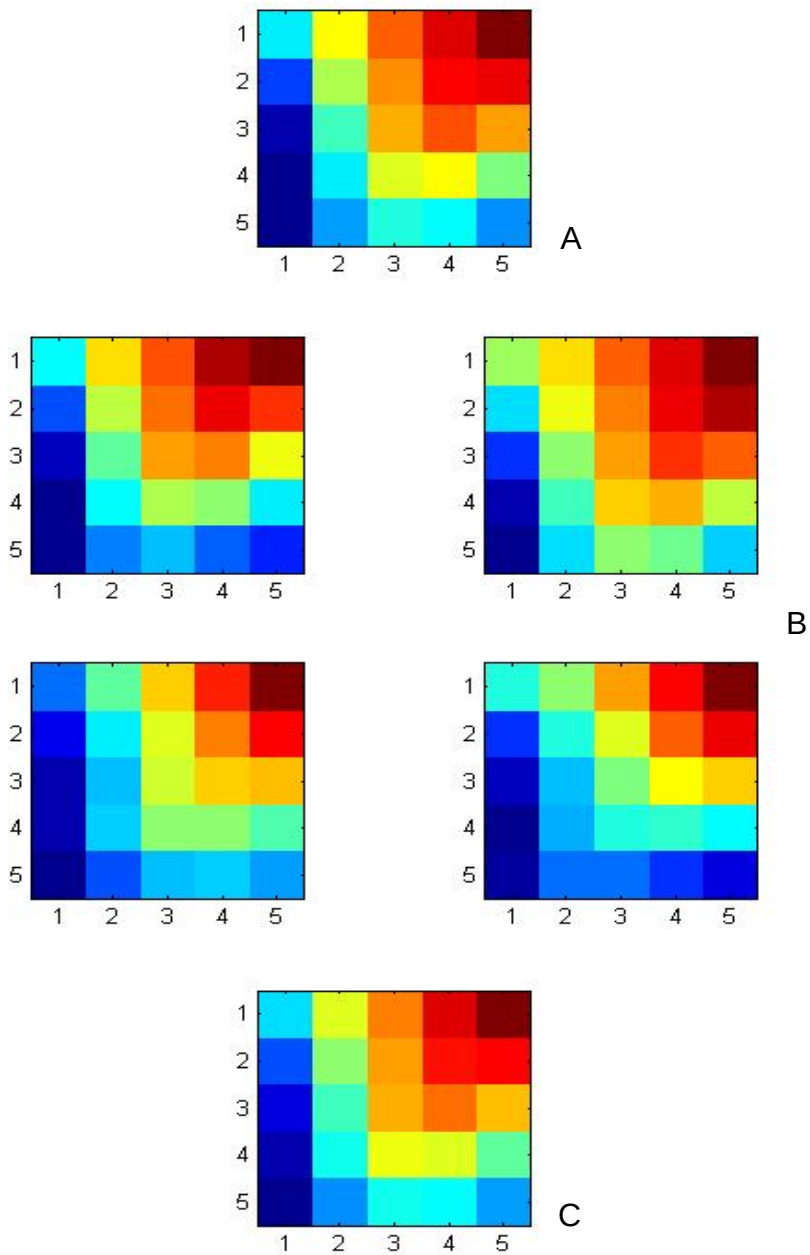


Figure 6-1. An example of nearest neighborhood searching algorithm applied to 5×5 low-resolution patch for 4 times magnification: A) input low resolution patch, B) 4 nearest neighbor patches extracted from the training images, C) reconstructed input patch by weighted combination of image patches in (B). (Weighting vector used: Γ_n={ 0.4787, 0.3110, 0.4518,-0.2415}). Darker red represents higher elevation.

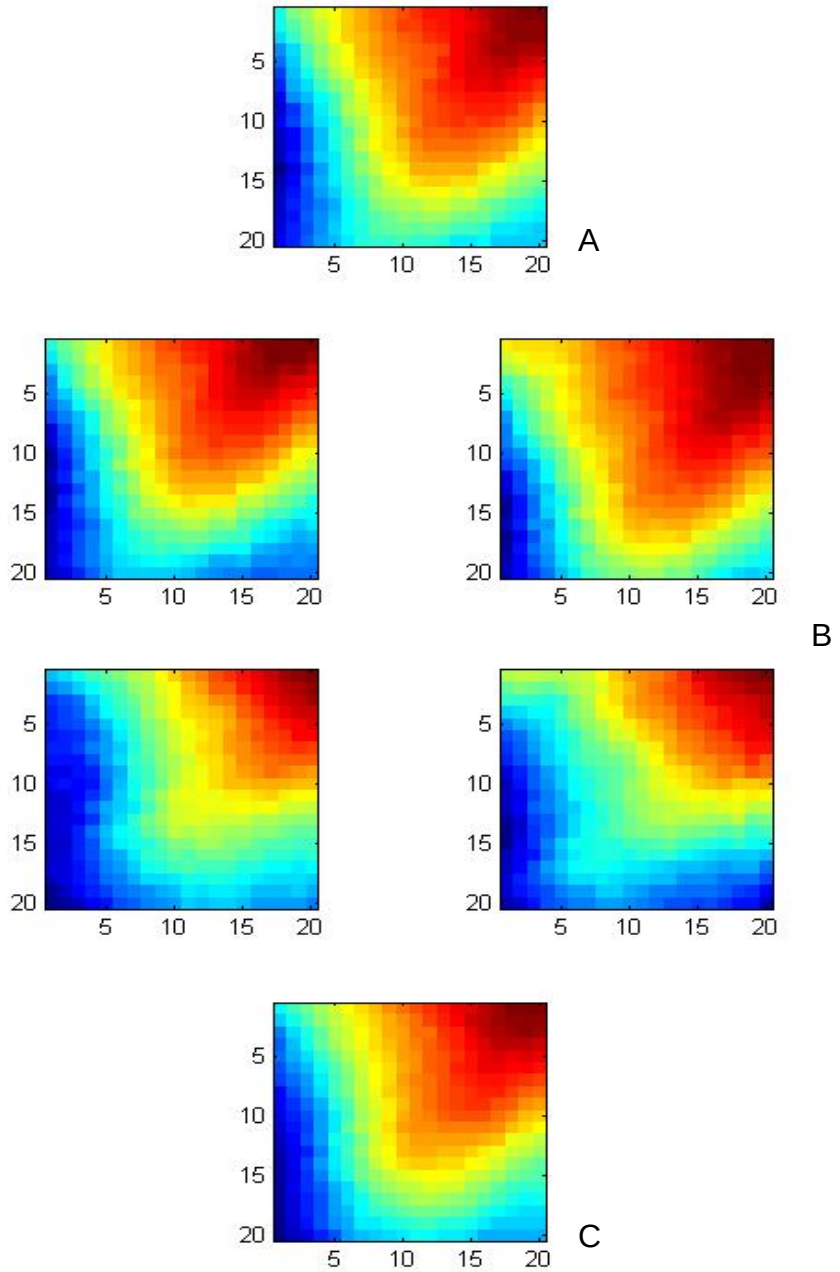


Figure 6-2. A) Corresponding high resolution patch of Figure 6-1 (A), B) 4 high resolution patches corresponding to low resolution patches in (A), C) estimated high resolution patch constructed by (B). (Weighting vector used: Γ_n={ 0.4787, 0.3110, 0.4518,-0.2415}) Darker red represents higher elevation.

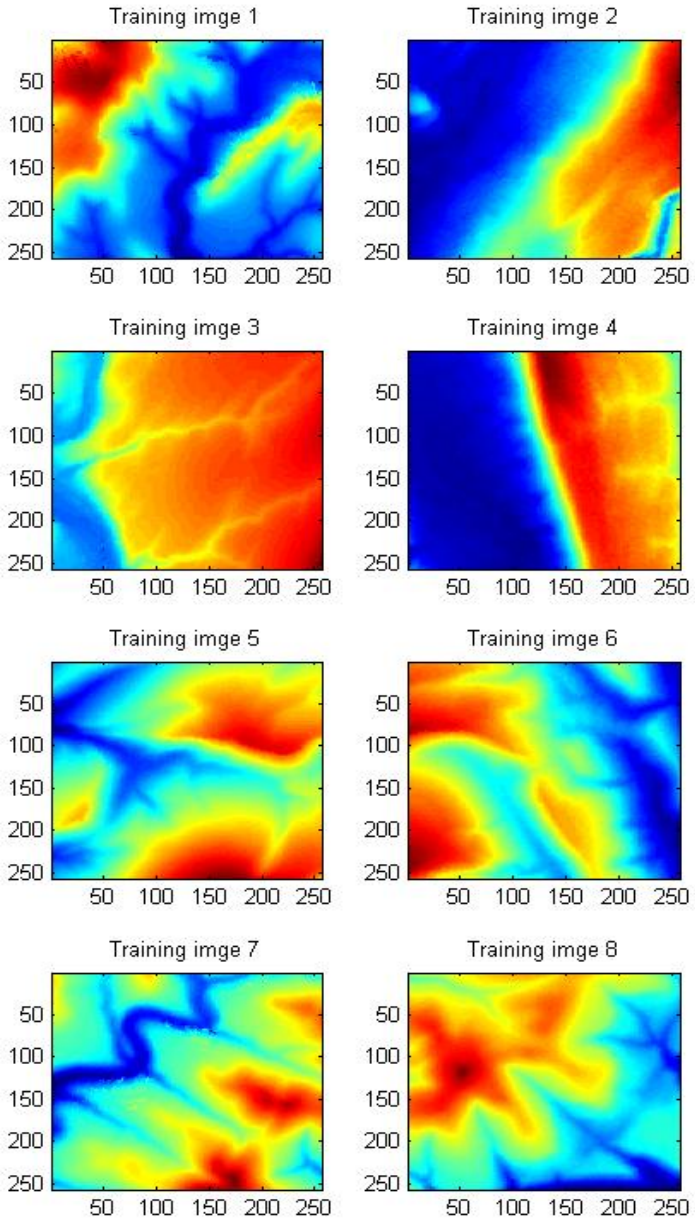


Figure 6-3. 8 sets of training images used for the results in Figure 6-1 and Figure 6-2. Only high resolution training images are shown. Each of corresponding low resolution training images is obtained by down sampling by order of 4.

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}

Figure 6-4. A 5×5 local neighborhood in the low-resolution image for computing the first-order and second-order gradients of the pixel at the center with elevation value x_{33}

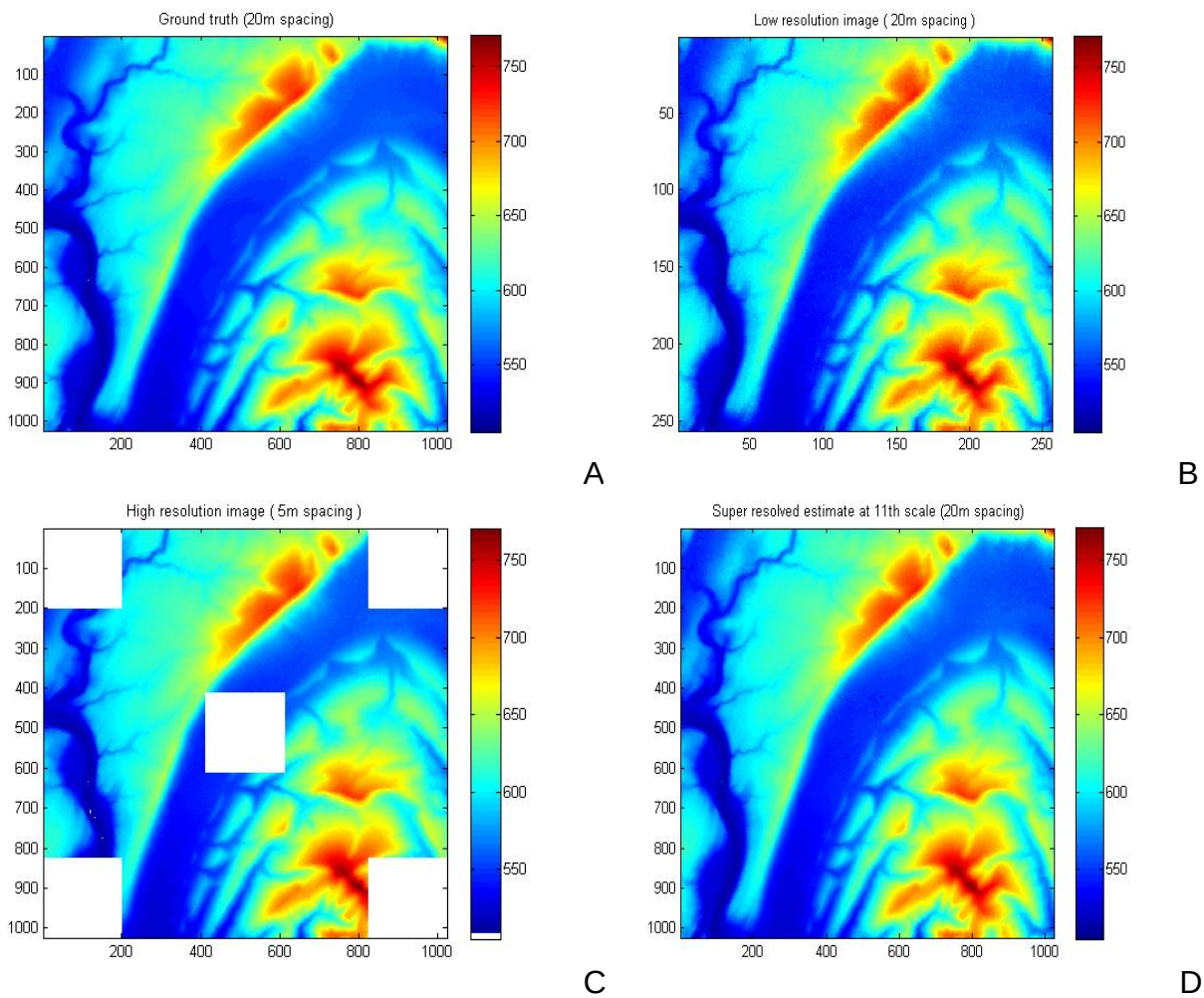


Figure 6-5 The input image sets to the fusion algorithm. A) Ground truth, B) low resolution image (256×256), C) sparse high resolution image (1024×1024: data void regions are represented as white), D) Super resolved image of smoothed estimate at coarse scale (at 9th scale). All elevation units are in meters.

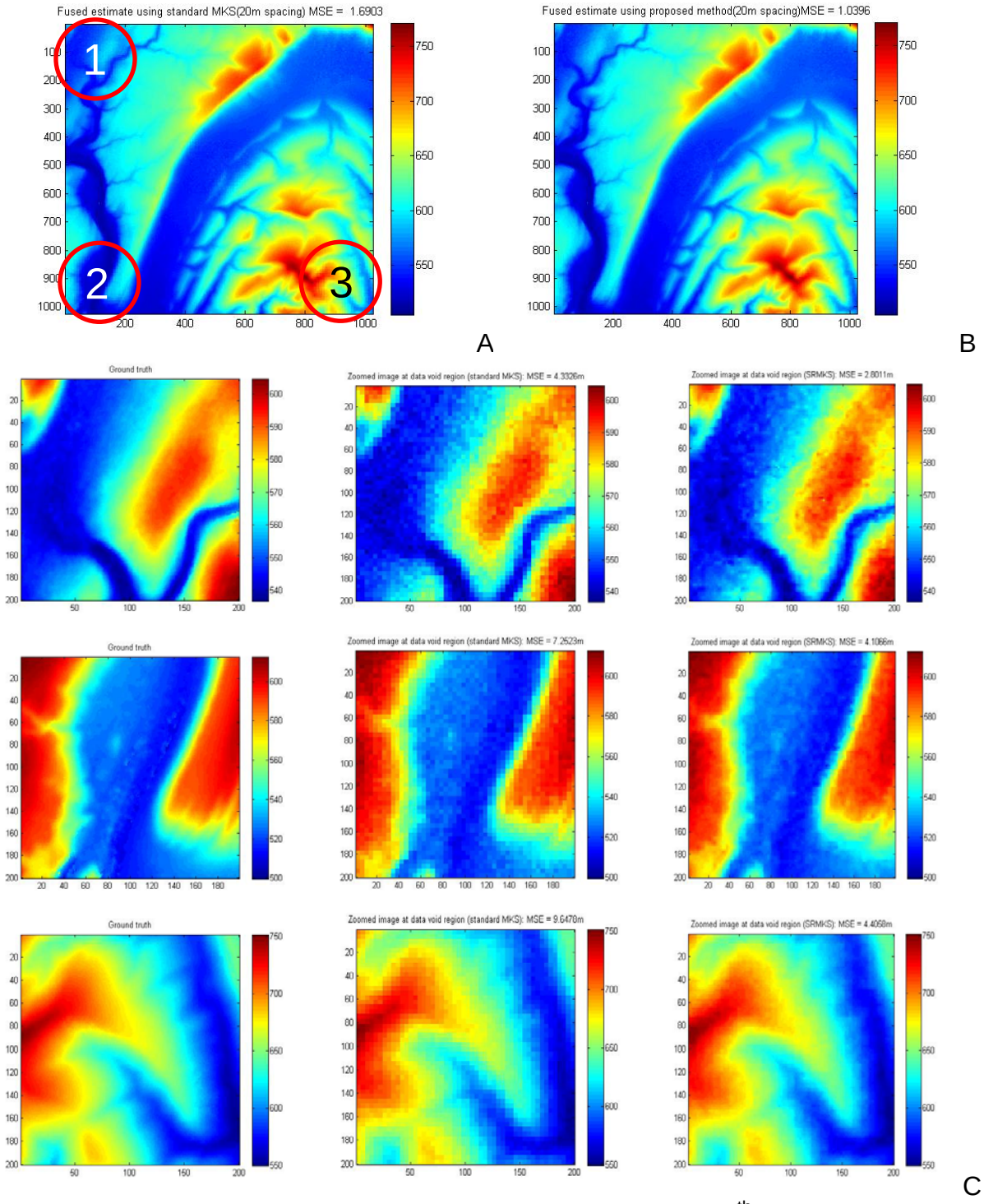


Figure 6-6 Comparisons of fused estimates at the finest scale (11th scale). A) using standard MKS introduced in Chapter 3, B) using proposed SR-MKS in Chapter 6, C) zoomed fusion results of data void areas (areas circled by red in (A)). 1st row: zoomed area at 1, 2nd row: zoomed area at 2, and 3rd row zoomed area at 3. For each row of result (left) fine-scale ground truth, (center) fused by MKS in Chapter 3, and (right) fused by proposed method (SR-MKS). All elevation units are in meters.

Table 6-1. Mean square errors of image fusion results in Figure 6-6 (in meters)

	Multi-resolution fusion algorithms	
	Using MKS in Chapter 3	Using SR-MKS in Chapter 6
Figure 6-6 (A) and (B)	1.6903	1.0396
Area #1 in Figure 6-6 (A)	4.3326	2.8011
Area #2 in Figure 6-6 (A)	7.2523	4.1066
Area #3 in Figure 6-6 (A)	9.6478	4.4058

CHAPTER 7
CONCLUSIONS AND FUTUREWORKS

7.1 Conclusions

This dissertation aims at developing and investigating various multi-resolution image fusion problems. Therefore, the robust image processing framework is suggested and employed to merge complementary multi-resolution images at desired resolutions. Combining of information acquired by different sources plays an important role in many research areas such as the Earth science for modeling dynamic environmental processes or some of important phenomenon that impact specific topography zone over a large range of spatial and temporal scales. By conducting comprehensive theoretical reviews and practical simulations, we become to know that Kalman filter based multi-resolution image fusion scheme successfully merges different sources of image sets. For these processes, we have employed statistically optimal fusion method utilizing multi-scale estimation framework (in Chapter 2 and Chapter 3) which provides both extensive coverage and locally high-resolution details in fused estimates. Using hierarchical image model on quadtree, we proved that highly parallelizable, thus efficient inference algorithm can be implemented for fusion process. Specifically, modeling of natural process such as topography or bathymetry is well-suited on multi-scale tree structure because it effectively captures multi-scale characters of natural processes or signals. By transforming spatial data models to hierarchical method, the Multi-scale Kalman Smoothing algorithm (MKS) is implemented to blend the information contained in images at different resolutions. This multi-scale approach is theoretically sound and has shown nice performance for estimation of image data in terms of computational complexity and optimality in mean square error sense. Unlike other

classical image inference methods such as traditional Markov Random Field (MRF) based, this is non-iterative to compute not only optimally fused estimates but error measures at different spatial scales which are useful for evaluating accuracy and confidence of algorithm under study. To show the potentials of proposed fusion scheme, we appropriately redesign algorithm to provide solutions for the cases when computational loads are too intense to be afforded by conventional MKS algorithm (in Chapter 4). The first case is arising when the image data sets being fused are differed in resolution by large order of scale and the fine-scale image available is spare (in section 4.1). If this is the case, the original form of MKS in Chapter 3 is inefficient. We aggregate this problem by reducing the floating point operations per node by deriving a new algorithm. It then reduced the number of nodes that must be processed depending on the degree of sparseness in finest-scale data, which led to a favorable reduction in floating point operations required for estimating a fused DEM from the component data. (more than 80% for data used in simulation). Also, we have proposed a new method to address vector formatted measurement fusion case (in section 4.2). In this work, we introduce an improved algorithm to fuse space-borne data from ERS-1/2 platforms with multiple sets of airborne data acquired by the NASA/JPL TOPSAR platform to obtain statistically optimal high-resolution estimates of topography. Because multiple measurements are available at the same scale from the same time period, simply iterating a scalar-valued MKS algorithm leads to the degenerative case of estimating an unknown constant, and the smoothed covariance is therefore driven asymptotically towards zero as a function of iteration. A vector-valued implementation of the MKS algorithm can in principle solve this problem, but the dimension of measurement being

processed grows overwhelmingly. We overcome this by processing the measurements sequentially, such that the Kalman gain, estimate covariance, and the a posteriori estimate are calculated serially using the information form of the MKS algorithm. We ameliorate this problem by employing a serial measurement update.

Despite of the attractive features driven by multi-scale based image processing, the primary disadvantage involved is the presence of pixel blockiness in final results. It is tradeoff between simplicity and accuracy, but it often makes ones hesitating to apply it in certain applications. As shown in Chapter 5 and Chapter 6, the blocky artifact is the natural phenomenon caused by non-loopy quadtree model and the algorithm implemented on this structure. The quadtree based model employed may not adequately model the correlation functions between spatially adjacent pixels. This leads quadtree structure producing distract blocky artifact (in Chapter 5). Therefore, in Chapter 6, Super Resolution Multi-scale Kalman Smoothing (SR-MKS) algorithm is suggested to address this problem. This novel method exploits the concepts of image super resolution technique. Based on the works in Chapter 3 and Chapter 4, an algorithm is developed to mitigate blocky artifact exhibited at data void regions of the finest scale in MKS algorithm. Single frame learning based image super resolution technique is employed in multi-resolution fusion scenario. The super resolved image is served as an auxiliary input to the vector-valued MKS such that it supports the regions where measurements are not available. This effectively suppresses the blockiness in estimates and returns both visually and analytically, thus theoretically acceptable, enhanced results.

Throughout this image fusion research, we have proved that multi-scale method provides less computational complexity, more accurate and flexible so that it is conveniently applicable on variety of real world remote sensing applications. Our main concerns of this dissertation have been much concentrated on the fusion of multi-resolution images, but it can be broadly extensible to other image processing applications such as image compression, pattern classification/recognition or, object detection.

7.2 Future Works

Although Image data sets considered in this dissertation have same modalities (e.g. elevation values or intensities), the fusion of different types of measurements can be also under consideration. For example, the multi-scale fusion of degraded or damaged color image with high resolution intensity image, or fusion of low luminance image with high resolution image which contains only curvature or edge information of interests.

These works will require more sophisticated data models because the characteristics of multi-resolution data sets being fused are completely heterogeneous. Unlike fusion of multi-resolution images having similar modality, more challenging steps must be preceded to understand the relations and compatibilities between image data sets. Once the appropriate modeling for multiple images is made, the fusion algorithms suggested in this dissertation would be redesigned or replaced to be suited with newly introduced models. These applications will be very useful in real industry applications such as designing low complexity mobile phone camera which supports high definition, vehicle navigation system driven by various sensors, as well as military or meteorological purposes. The impact of efficient and reliable information fusion will be

appreciated by various fields of remote sensing researches because the costly data measuring procedures can be replaced by effective and convenient signal processing techniques.

LIST OF REFERENCES

N.J. Adams. Dynamic Trees: A Hierarchical Probabilistic Approach to Image Modeling, *PhD dissertation*, Division of Informatics, Univ. of Edinburgh, Edinburgh, U.K., 2001.

S. Baker and T. Kanade. Hallucinating Faces, *Fourth International Conference on Automatic Face and Gesture Recognition*, 2000.

J. Besag. Spatial interaction and the statistical analysis of lattice systems, *J. Royal Statistical Society B,* vol. 36, pp. 192-225, 1974.

C.M. Bishop, A. Blake, and B. Marthi. Super-resolution enhancement of video, Int. *Conf on Artificial Intelligent Statistics*, Key West, FL, 2003.

C.A. Bouman, and M. Shapiro. A multiscale random field model for Bayesian image segmentation, *IEEE Transactions on Image Processing*, 3, 162–177, 1994.

R. Brown, and P. Hwang. *Introduction to Random Signals and Applied Kalman Filtering,* Wiley, New York, NY, 3rd ed., 1997.

W.E. Carter, R.L. Shrestha, G. Tuell, D. Bloomquist, and M. Sartori. Airborne Laser Swath Mapping Shines New Light on Earth's Topography, *EOS, Transactions American Geophysical Union*, 82(46), 549-555, 2001.

H. Chang, D.Y. Yeung, and Y. Xiong. Super-Resolution Through Neighbor Embedding, *Proc. of IEEE Conf. CVPR*, 2004.

M.J. Choi, and A.S. Willsky. Multiscale Gaussian Graphical Models and Algorithms for Large-Scale Inference, *IEEE 2007 Statistical Signal Processing Workshop*, Madison, Wisconsin, Aug. 2007.

K.C. Chou. A stochastic modeling approach to multiscale signal processing, *Ph.D. dissertation*, Dept. Elec. Eng. and Comp. Sci., Massachusetts Institute of Technology, Cambridge, MA, May 1991.

K. Chou, A. Willsky, and R. Nikoukhah. Multiscale systems, Kalman filters, and Riccati equations, *IEEE Transactions on Automatic Control* 39 3, pp. 479–492. 1994.

K. Chou, A.S. Willsky, and A. Benveniste. Multiscale recursive estimation, data fusion and regularization, *IEEE Trans. Automat. Contr.*,vol.39, pp. 464-478, 1994.

M. Daniel, and A. Willsky. A multiresolution methodology for signal-level fusion and data assimilation with applications to remote sensing, *Proc. IEEE*, vol. 85, pp. 164–180, Jan. 1997.

P.J.J. Desmet, and G. Govers. Comparison of routing algorithms for digital elevation models and their implications for predicting ephemeral gullies, *International Geographical Information Systems*, 10, pp.311-331, 1996.

R.O. Duda, P.E. Hart and D.G. Stork. *Pattern Classification,* Wiley-Interscience, New York, NY,2001.

P.W. Fieguth, W. C. Karl, A. S. Willsky, and C. Wunsch. Multiresolution optimal interpolation and statistical analysis of TOPEX/POSEIDON satellite altimetry, *IEEE Trans. Geoscience and Remote Sensing*, vol. 33, pp. 280–292, Mar. 1995.

P.W. Fieguth, W. Irving, and A. Willsky. Multiresolution Model Development for Overlapping Trees via Canonical Correlation Analysis, *Proc. IEEE International Conference on Image Processing* (I), pp.45-48, 1995.

W.T. Freeman, T.R. Jones, and E.C. Pasztor. Examplebased superresolution, *IEEE Computer Graphics and Applications*, 2002.

W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision, *International Journal of Computer Vision*, 2000.

J.C. Gallant, and M.F. Hutchinson. Producing digital elevation models with uncertainty estimates using a multi-scale Kalman filter, *7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences.,* pp.150~159, 2006.

Q. Gan, and C.J. Harris. Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion, *IEEE Transactions on Aerospace and Electronic System*, 37, issue 1, pp. 273-279, 2001.

D. Gesch, J. William, and W. Miller. A comparison of U.S. geological survey seamless elevation models with shuttle radar topography mission data, *Geoscience and Remote Sensing Symposium. IEEE 2001 International*, 2, pp. 754-756, 9-13 July, 2001.

R. Gonzalez, and R. Wood. *Digital Image Processing*, Prentice Hall, 2nd ed, Upper Saddle River, NJ ,2002.

H. Greenspan, C. Anderson, and S. Akber. Image enhancement by nonlinear extrapolation in frequency space, *IEEE Transactions on Image Processing*, 9(6), 2000.

M. Grewal, and A. Andrew. *Kalman Filtering: Theory and Practice*, Englewood Cliffs, New jersey, 1993.

D. Hall, and J. Linas. An introduction to multisensor data fusion, In *Proceedings of the IEEE*, 85, pp. 6-23., Jan, 1997.

S. Haykin. *Adaptive Filter Theory*, Prentice Hall, 4thed, Upper Saddle River, NJ, 2002.

IHRC. WSMP (Windstorm Simulation & Modeling Project): Airborne LiDAR data and digital elevation models in Miami-Dade, Florida, final report. International Hurricane Research Center (IHRC), 2004.

M. Irani, and S. Peleg. Improving resolution by image registration, *CVGIP:Graphical Models and Image Process* 53, 231–239, 1991

W. Irving, W. Karl, and A. Willsky. A Theory for Multiscale Stochastic Realization, *33rd Conference on Decision and Control,* Orlando, FL, pp.655-662, 1994.

W. Irving. Multiresolution Stochastic Realization and Model Identification with Applications to Large-Scale Estimation Problems, *PhD thesis*, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Sept. 1995.

H. Jhee, S. Cheung, and K. Clint. Slatton. Efficient Observational Updating for Fusion of Incomplete Image Data, *Proc. IEEE 2005 International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 4, pp. 2807 – 2810. , Jul., 2005.

R.A. Johnson, and D. W. Wichern. *Applied Multivariate Statistical Analysis*, Prentice Hall, Upper Saddle River, NJ, 4th ed., 1998.

M.V. Joshi and S. Chaudhuri. A learning-based method for image superresolution from zoomed observations, *Proc Fifth Int Conf. Adv Pattern Recogn*, Indian Statistical Institute, Kolkata, India, pp. 179–182, 2003

E.W. Kamen, and J. K. Su. *Introduction to Optimal Estimation,* Springer-Verlag, London, U.K, 1999

S.M Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, Upper Saddle River, NJ ,1993.

S.P. Kim, N.K. Bose, and H.M. Valenzuela. Recursive reconstruction of high resolutionimage from noisy undersampled multiframes, *IEEE Trans Accoustics,Speech, Signal Process* 38, 1013–1027, 1990

P.D. Komar. *Beach Processes and Sedimentation*, Prentice Hall, Upper Saddle River, NJ, 2nd ed. 1998.

P. Kumar. A multiple scale state-space model for characterizing subgrid scale variability of near-surface soil moisture, *IEEE Trans. Geosci. Remote Sensing*, vol. 37, pp. 182–197, Jan. 1999.

J. Laferte, P. Perez, and F. Heitz. Discrete Markov image modeling and inference on the quadtree, *IEEE Trans. Image Proc.*, vol. 9, no. 3, pp. 390–404, 2000.

M.R. Luettgen, W.C. Karl, and A. Willsky. Multiscale Representations of Markov Random Fields, *IEEE Trans. Signal Processing*, vol. 41, pp. 3377-3395, 1993.

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

A. Mário, T. Figueiredo, and R.D. Nowak. An EM algorithm for wavelet-based image restoration, *IEEE Transactions on Image Processing*, 2003.

B. Morse, and D. Schwartzwald. Image magnification using level set reconstruction, *Proc. of Int'l Conf. on Computer Vision*, pp. 333-341, 2001.

K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference: an empirical study, *Proceedings of Uncertainty in AI*, 1999.

K. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning, *PhD thesis*, Univ. of California, Berkeley, 2002.

NGDC. National Geophysical Data Center (NGDC), Available online at: www.ngdc.noaa.gov, National Oceanic & Atmospheric Administration, Washington, DC. 2005.

NOAA, National Geophysical Data Center (NGDC), www.ngdc.noaa.gov

B. Potamianos, and J . Goutsias. Stochastic simulation techniques for partition function approximation of Gibb's random field images, *Technical rept*., Dep. Electrical and Computer Engineering, Johns Hopkins University. Baltimore, MD., 1. Jan 1990 – 30. Jun 1991.

M.K. Schneider, P. W. Fieguth, W. C. Karl, and A. S. Willsky. Multiscale methods for the segmentation and reconstruction of signals and images, *IEEE Trans. Image Proc.*, vol. 9, pp. 456–468, Mar. 2000.

K.C. Slatton, S. Cheung, and H. Jhee. Reduced-complexity fusion of multiscale topography and bathymetry data over the Florida coast, *IEEE Geoscience and Remote Sensing Letters*, 2, pp. 389-393., 2005.

K.C. Slatton, M. M. Crawford, and B. L. Evans. Combining interferometric radar and laser altimeter data to improve estimates of topography, *in Proc.IEEE Int. Geosci. Remote Sensing Symp.*, vol. 3, (Honolulu, HI), pp. 960– 962, Jul. 2000.

K.C. Slatton, M.M. Crawford, and B.L. Evans. Fusing interferometric radar and laser altimeter data to estimate surface topography and vegetation heights, *IEEE Transactions on Geoscience and Remote Sensing*, 39, pp. 2470-2482., 2001.

H.W. Sorenson. Least-square estimation: From Gauss to Kalman, *IEEE Spectrum*, pp.63-68, 1970.

J.L. Starck, F. Murtagh, and A. Bijaoui. *Image Processing and Data Analysis: the Multiscale Approach*, Cambridge University Press, Cambridge, UK, 1998.

G. Strang. Wavelets and dilation equations: A brief introduction., *SIAM Rev*. Volume 31, Issue 4, pp. 614-627, Dec. 1989.

R.Y. Tsai, and T.S. Huang. Multiframe image restoration and registration, In Advances in computer vision and image processing, *JAI Press*, pp.317–339. 1984

R. Tenney, and A. Willsky. Multiresolution estimation for image processing and fusion, *Proc. AFIT Workshop on Wavelets and Signal Processing,* 1992

S. Todorovic, and M.C. Nechyba. Interpretation of complex scenes using dynamic tree-structure Bayesian networks, *Computer Vision and Image Understanding*, v.106 n.1, p.71-84, Apr. 2007.

D.L. Turcotte. *Fractals and Chaos in Geology and Geophysics*, Cambridge University Press*,* 2nd ed.1997

A.S. Willsky. Multiresolution Markov models for signal and image processing, *Proceedings of the IEEE 90 (8),* 1396–1458., 2002

K. Zhang, S.C. Chen, D. Whitman, M. Shyu, J. Yan, and C. Zhang. A progressive morphological filter for removing non-ground measurements from airborne LiDAR data, *IEEE Transactions on Geoscience and Remote Sensing*, 41, pp. 872-882.,2003.

S. Zhao, H. Han, and S. Peng. Wavelet-domain HMT-based image superresolution, *Proc. of IEEE Int'l Conf. on Image Processing*, 2003.

H. Zebker, S. Madsen, and J, Martin. The TOPSAR Interferometric Radar TopographicMapping Instrument, *IEEE Trans. Geosci. Remote Sensing*, vol. 30, NO.5, pp. 933–940, Sep 1992.

H.A. Zebker, T. G. Farr, R. P. Salazar, and T. H. Dixon. Mapping the world's topography using radar interferometry: The TOPSAR mission, *Proc. IEEE*, vol. 82, pp. 1774–1786, Dec. 1994.

BIOGRAPHICAL SKETCH

Hojin Jhee was born in Seoul, South Korea. He received Bachelor of Science degree in electronic engineering at Dongguk University, Seoul, South Korea in 1997 and his Master of Science in electrical and computer engineering at University of Florida, Gainesville, Florida, U.S.A. in 2001.

At University of Florida, he was a member of Adaptive Signal Processing Laboratory (ASPL) under Dr. Clint Slatton until August, 2008, and has served as a research assistant at High Speed Digital Architecture Laboratory (HSDAL) from August 2008 to May 2010. He received his Ph.D. degree in electrical and computer engineering at the University of Florida May 2010. His research interests include statistical signal processing, data fusion, digital image processing, machine learning and remote sensing.