Hierarchical Stochastic Modeling for Multiscale Segmentation and Compression of SAR Imagery

by

Andrew J. Kim

B.E.E., Georgia Institute of Technology (1995)

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

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1997

Author .....
Department of Electrical Engineering and Computer Science
May 16, 1997

Certified by .. ..
Hamid Krim
Research Scientist, Laboratory for Information and Decision Systems
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Hierarchical Stochastic Modeling for Multiscale Segmentation and Compression of SAR Imagery

by

Andrew J. Kim

Submitted to the Department of Electrical Engineering and Computer Science
on May 16, 1997, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Abstract

There has recently been a growing interest in Synthetic Aperture Radar (SAR) imaging on account of its importance in a variety of applications. One attribute leading to its gain in popularity is its ability to image terrain at extraordinarily high rates. Acquiring data at such rates, however, has drawbacks in the form of exorbitant costs in data storage and transmission over relatively slow channels, and addressing these problems is clearly important. To abate these and related costs, we propose a segmentation driven compression technique using hierarchical stochastic modeling within a multiscale framework. Our approach to SAR image compression is unique in that we exploit the multiscale stochastic structure inherent in SAR imagery. This structure is captured by a set of stochastic models which accurately characterize the evolution in scale of homogeneous regions of different classes of terrain. We thus use them to generate a multiresolution segmentation of the image which is subsequently used in tandem with the corresponding models in a pyramid encoder to provide a robust, hierarchical compression technique. In addition to coding the segmentation, this technique achieves high compression ratios with remarkable image quality.

Thesis Supervisor: Hamid Krim
Title: Research Scientist, Laboratory for Information and Decision Systems

Acknowledgments

There are many people I would like to thank for the help and support they have given me these past few years. First and foremost, I would like to thank my thesis advisor Hamid Krim for his immeasurable guidance and support over the last two years. This work would not have been possible without the direction that he provided, nor would it have been as enjoyable without his contagious enthusiasm. I am also greatly indebted to Alan Willsky who was another boundless source of ideas, insight, and direction. I would also like to express my gratitude toward Clem Karl and Jun Zhang who provided a great deal of food for thought in our weekly meetings.

I am grateful for my friends at MIT, in addition to those above, who have helped me both academically and personally. In particular, Nick "Days of Our Lives" Laneman, Walter "SFECMD"[1] Sun, and Dewey "nap time" Tucker with whom I have been close friends my entire time here. There are also those SSG members who were always willing to discuss ideas and help me out: Mike "PAC–10" Daniel, Paul "manifesto" Fieguth, Charlie "Big Dog" Fosgate, Austin "all sport" Frakt, Ben "the Godfather" Halpern, Terrence "the puppet president" Ho, Seema "MTV" Jaggi, Rachel "no accent" Learned, Cedric "da' MAN" Logan, Ilya "chocolate" Pollak, John "John John" Richards, Mike "the juggler" Schneider, Andy "`e.r.`" Tsai, and Dewart "hard liquor" Tucker.

Finally, I would like to thank my family to whom I owe so much. My mother and father for supporting and believing in me; Joe, Ceci, Mike, Tina, and Pat for their care and advice; & Shelby for helping me understand which things in life are truly important. Thank you.

[1] Super Fly East Coast Mac Daddy

This work was sponsored in part by Air Force Office of Scientific Research (BU GC12391NGD) and in part by the Advanced Research Project Agency (F49620-93-1-0604).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, synthetic aperture radar (SAR) imaging has been rapidly gaining prominence in applications such as remote sensing, surface surveillance, and automatic target recognition (ATR). Its growing popularity is largely due to its remarkable ability to rapidly image vast areas of terrain at a fine resolution. In fact, the coverage rates of an airborne SAR system are capable of exceeding $1km^2/s$ at a resolution of $1m^2$, thus producing over one million pixels per second. This daunting amount of acquired data necessitates either a large capacity transmission link to a ground station or an excessively large memory for storage. In addition to the technical challenges posed by such data, the associated costs of its handling together with the physical limitations of an on board SAR make it imperative to investigate ways of abating such impediments. Previous efforts in this area have focused on compressing different representations of the SAR data including the raw SAR data [1], the complex SAR image, and the log–magnitude SAR image with the particular compression modality largely determined by its intended application. Our technique is specifically aimed at compressing the log–magnitude image which can be reliably used for target detection

and recognition. Other algorithms used to compress this real–valued image include JPEG and various transform–based techniques such as the Gabor transform, wavelet transform, and best–basis wavelet packet transform [2]. We present in this thesis a new compression technique which differs from these, in that it takes great advantage of a multiscale framework that provides a preliminary analysis in the form of a segmentation map. The latter is obtained by using the stochastic properties of prevalent classes of terrain within the imagery.

The approach taken in this thesis carries out a partial analysis of SAR imagery and with minimal loss, proceeds to efficiently transform the data for transmission over a low capacity channel. A SAR image multiscale representation that results from tree–based hierarchical stochastic modeling is used for segmenting and subsequently compressing the imagery. In this multiscale framework, scale autoregressive (AR) models effectively capture the evolution in scale of homogeneous regions of terrain. We thus associate with each major classification of terrain, a predefined model characterizing its evolution in scale. We obtain a segmentation of SAR imagery by comparing the local evolution in scale of the imagery to predefined models. The algorithm is sufficiently simple to be performed onboard the SAR imaging system, and thus eliminating the need to transmit the entire complex SAR image to the ground station in cases where the only use of the phase information is for segmentation. One such instance is in ATR where the log–magnitude image produces statistically reliable results for target detection. The performance of target detection algorithms may be enhanced by incorporating terrain information supplied by the segmentation. This enhancement is in fact the primary motivation for obtaining a segmentation. Performing the segmentation on board the SAR system not only frees us of the burden of having to code the phase information, but also affords via the map, an efficient compression technique that takes advantage of the local dependencies specific to various classes of terrain. We thus use the segmentation in tandem with the corresponding

multiscale models, to obtain an efficient multiresolution compression technique that abates the costs associated with data storage / transmission.

1.1 Coherent Imaging

SAR is a coherent imaging device that uses radar to measure the reflectivity of terrain. Particularly, the magnitude m and phase ψ of the reflectivity of the terrain are measured by comparing a transmitted radar signal to the received echo. In representing the relationship between a transmitted signal and its corresponding return, it is useful to model the terrain as consisting of a finite set of discrete objects (scatterers). Considering the case of a single scatterer σ_0 as an example, the return from transmitting the signal $s(t)$ can be written as

$$g_0(t) = m_0 e^{j\psi_0} s(t - 2T_0),$$

where m_0 and ψ_0 are the gain and phase shift, respectively, associated with σ_0 and T_0 is the travel time of the signal from the SAR to σ_0. Each pixel in SAR imagery corresponds to a small patch of terrain (resolution cell) that can be thought of as being composed of a finite number of scatterers indexed by $i \in \mathcal{I}$. Because the size of a resolution cell is small relative to its distance from the SAR, by superposition, the received echo for this resolution cell can be approximated as

$$\begin{aligned} g(t) &= \sum_{i \in \mathcal{I}} g_i(t) \\ &= \sum_{i \in \mathcal{I}} m_i e^{j\psi_i} s(t - 2T_i) \\ &\approx s(t - 2T) \sum_{i \in \mathcal{I}} m_i e^{j\psi_i}, \end{aligned} \tag{1.1}$$

where m_i, ψ_i, and T_i are the gain, phase shift, and one–way travel time associated with scatterer σ_i and T is the travel time from the SAR to the center of the resolution cell. We thus have that each pixel in a SAR image represents a coherent average of the reflectivity of scatterers contained within its corresponding resolution cell.

As a coherent sensing device, SAR imagery exhibits a phenomenon known as speckle that produces a noise like appearance in the imagery. The origin of speckle can be understood from inspection of Eq. (1.1). As a complex sum, the returns from individual scatterers within a resolution cell will interfere both constructively and destructively. Smooth variations in the gain of the imagery may then appear as sharp gradients in the log–magnitude image due to this interference. Figure 1.1 illustrates an example of how objects with nearly equal gains can result in drastically different pixel values. This hypothetical case depicts how, depending on the phase, the returns from three scatterers with equal magnitude can superimpose to result in either a large or small magnitude return for the entire resolution cell.

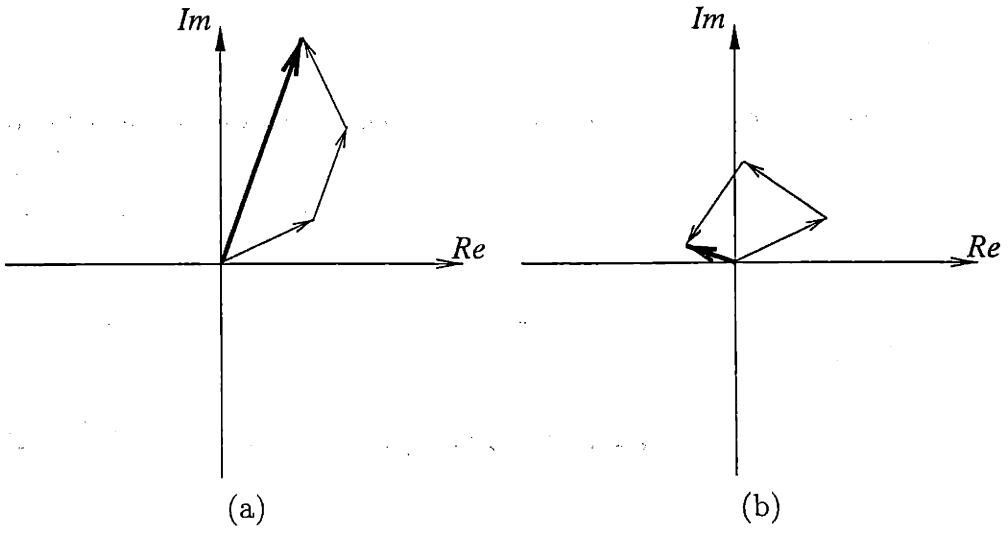


Figure 1.1: Illustration of how three equivalued radar returns could superimpose giving an overall return (bold vector) that is (a) large or (b) small in magnitude.

1.2 Multiscale Modeling

The interference among radar returns is one motivating factor for modeling multiresolution sequences of SAR imagery according to terrain. The idea is that the physical attributes of the different classes of terrain give rise to distinct multiresolution behavior. For example, grass terrain is composed of very many scatterers with small magnitude while forest terrain is composed of fewer scatterers with larger magnitude. One would thus expect there to be more correlation between pixels at different scales in the forest case since fewer random variables are involved. Experimentation has in fact shown this to be the case. To capture the behavior among the different resolutions of the image, we employ the multiscale framework introduced by Irving[3]. The generation of the multiresolution images and the scale AR models used to capture the evolution in scale are described in detail in Chapter 2.

1.3 Thesis Contributions

The work in this thesis represents an extension of the work done by Irving[3] and Fosgate[4]. Fosgate used the multiscale framework developed by Irving to segment SAR imagery according to terrain. Specifically, he classified each pixel as belonging to either a region of grass or of forest. We modify the residual–based approach taken by Fosgate to a model–based approach. The resulting segmentation is then used for compressing the log–magnitude SAR image.

1.3.1 Segmentation

A terrain segmentation of SAR imagery partitions the image according to the type of terrain to which each pixel belongs. Such a segmentation provides utility in many remote sensing applications. For example, in the context of ATR, segmentation can allow us to bypass processing of uninteresting imagery. It is known that high frequency radar is incapable of penetrating dense foliage, thereby making it impossible to detect targets lying under forest canopies. Knowledge of dense homogeneous forest regions would thus prevent futile analysis of the imagery for undetectable targets. In addition to its utility in image analysis, terrain segmentation also affords us a spatially adaptive image compression algorithm that takes advantage of having a statistical characterization for the terrain comprising the image.

The segmentation technique presented in this thesis builds upon the idea used by Fosgate, but uses a model–based approach instead of a residual–based approach. Doing so achieves greater statistical significance with the same window size. A high level description of our approach is given in Figure 1.2. To determine the classification of a SAR image pixel, we examine an observation window centered around that pixel at a succession of scales. Fosgate applied a maximum likelihood detector to the window of error residuals associated with predefined multiscale models. We instead find the best model describing the evolution in scale of the window and compare it to that of the predefined terrain classes using a maximum likelihood detector. Compared to its predecessor, our approach has shown to achieve considerably higher statistical significance thus enabling the use of smaller observation windows. The smaller window size increases the locality of the measurements thereby improving performance, particularly near terrain boundaries. In our technique, we further advance the ideas in [4] by applying this analysis to a subsampled set of pixels to obtain a subsampled image of sufficient statistics. The image of statistics is then interpolated to the full

sized image and thresholded to obtain the segmentation. This procedure bypasses much of the redundancy associated with overlapping windows, but does not suffer from any significant misclassifications or any blocking artifacts.

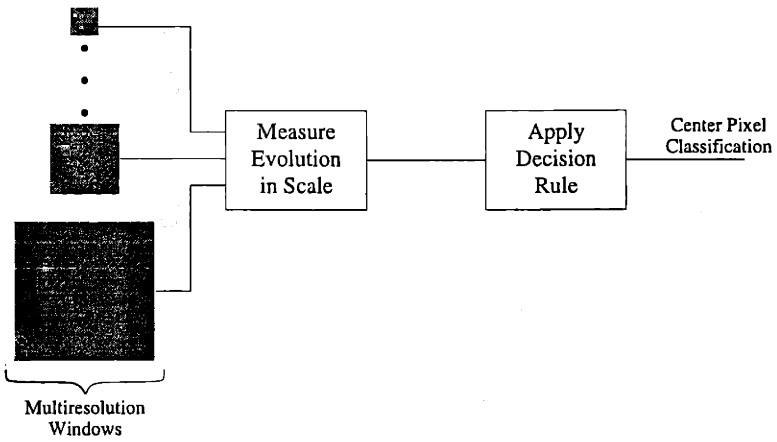


Figure 1.2: Structure of the segmentation algorithm. The evolution in scale of a sequence of small multiresolution windows is measured and subsequently used to classify the center pixel.

For the work in this thesis, we concentrate on the specific classifications of grass (or open fields) and forest due to their preponderance in SAR imagery. Even though we restrict ourselves to the binary case in order to simplify the presentation, this technique could be straightforwardly extended to multiple hypotheses through a sequence of binary decision rules.

1.3.2 Compression

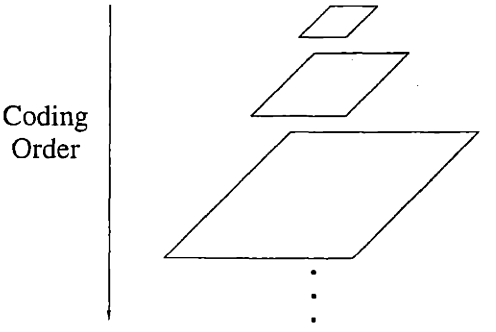
As stated at the beginning of this chapter, SAR systems rapidly acquire enormous amounts of data. The utility of SAR image compression is reflected by the potential increase in efficient data gathering capabilities and by the bandwidth reduction for

transmitting the data to a ground station. In addition to being applied on board the SAR system, compression could be used to compactly archive imagery.

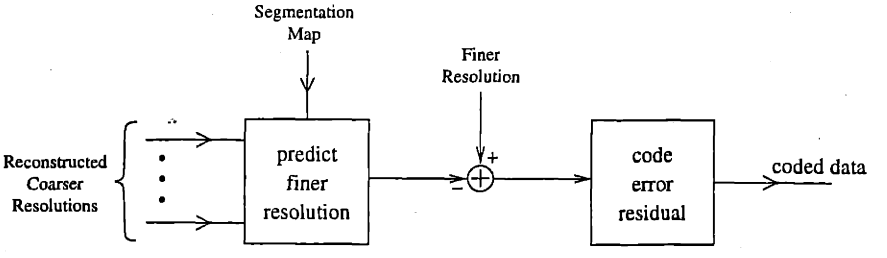
The efficacy of our compression technique is derived from the accurate modeling of terrain behavior within the multiscale framework. The terrain segmentation supplies knowledge of local dependencies within the imagery associated with the predetermined classes. To take advantage of this knowledge, we use the multiscale modeling within a pyramid–structured encoder that predicts fine resolution images using previously coded coarser resolutions. A high level diagram of this technique is given in Figure 1.3. We start by coding the segmentation map and coarsest resolution image. With the knowledge of the terrain comprising the image, we predict the next finer resolution using the appropriate multiscale model according to the segmentation map. To compensate for any fine details overlooked by the prediction, we then code the prediction error residual. We iterate this prediction and error coding process until the image is reconstructed at the full resolution. By following such a procedure, we eliminate much of the redundancy inherent in the various terrain classes to provide an efficient compression technique..

1.4 Thesis Organization

The remainder of this thesis is divided into four chapters. Chapter 2 provides a background describing the fundamental tools used throughout this thesis. In Chapter 3, we present our approach to SAR image segmentation using a statistical characterization of the evolution in scale, while in Chapter 4, we describe its application together with the multiscale framework to efficiently compress the imagery. Chapters 3 and 4 both demonstrate the effectiveness of the techniques presented therein with applica-



(a) Illustration of how the coding hierarchy proceeds from coarse to fine resolution images.



(b) Procedure used to code finer resolution images utilizing knowledge of terrain segmentation and coarser resolution images.

Figure 1.3: The Structure used for hierarchical coding of the multiresolution images.

tions to actual SAR data. We then conclude with Chapter 5 by providing conclusions and extensions of this research for possible future work.

Chapter 2

Background

While the formal approach to image analysis is perhaps invariant or universal to all images, the diversity and source of the images have a direct impact on the modeling and thus on the specific technique selected for the task. In this chapter, we provide some fundamentals relevant to the topics discussed in this thesis. In Section 2.1, we discuss the basics of stripmap SAR image formation which is followed by Section 2.2 where we give a description of an adapted multiscale analysis framework based on quadtree hierarchical stochastic modeling. In Section 2.3, we provide an overview of hypothesis testing paying particular attention to the maximum likelihood (ML) detector for Gaussian densities. In Section 2.4, we define the Shannon entropy and discuss its application in data compression. We conclude with a brief introduction to wavelets in Section 2.5.

2.1 SAR Image Formation

Stripmap SAR is a microwave radar imaging system capable of imaging terrain and targets at a fine resolution by artificially synthesizing a large aperture antenna. Advantages of using microwave radar over optical imaging techniques include the ability to penetrate cloud cover, to image terrain at night, and to operate at a resolution that is independent of altitude. Microwave imaging systems are at a disadvantage, however, in that they require an impractically long antenna to produce the narrow beamwidth necessary to achieve the same imaging resolution as optical systems. SAR systems circumvent this limitation by emulating a sufficiently long antenna by obtaining multiple measurements (typically thousands) of the terrain's reflectivity as the SAR platform (usually an airborne or spaceborne craft) moves across the terrain. The multiple returns are then coherently processed as measurements from an array of sensors phased as a beamformer antenna, thus providing a high imaging resolution. Stripmap and spotlight are the two main modes of SAR and differ slightly in their acquisition of data. For stripmap SAR, the orientation of the antenna is fixed with respect to the platform, while for spotlight SAR, the position of the antenna is continually adjusted to constantly illuminate a particular patch of terrain. Spotlight SAR is thus suited for the close examination of a relatively small region, whereas stripmap SAR is used to image large swathes of terrain that can exceed several kilometers in length. Data acquisition rates are more excessive for the former and present a major challenge in the excessive costs of data transmission/storage. We thus specifically focus on stripmap SAR.

We start by describing the basics of a SAR data acquisition system as shown in Figure 2.1. The coordinate in the direction of the flight path of the SAR platform is the *cross range* (*azimuth*) while the coordinate in the direction of the radar beam

(i.e. that of the transmitted radar pulse) is the *slant range*. The projection of the slant range onto the terrain is the *range*. The *nadir* is the point vertically below the SAR platform in the range–cross range plane. The terrain within the half-power beamwidth of the antenna is the *footprint* and represents the area illuminated by a single radar pulse. The strip of terrain illuminated throughout the flight path is the *swath*. The information in each radar pulse return represents the information in the slant range–cross range plane which can be converted to the range–cross range plane via simple trigonometry.

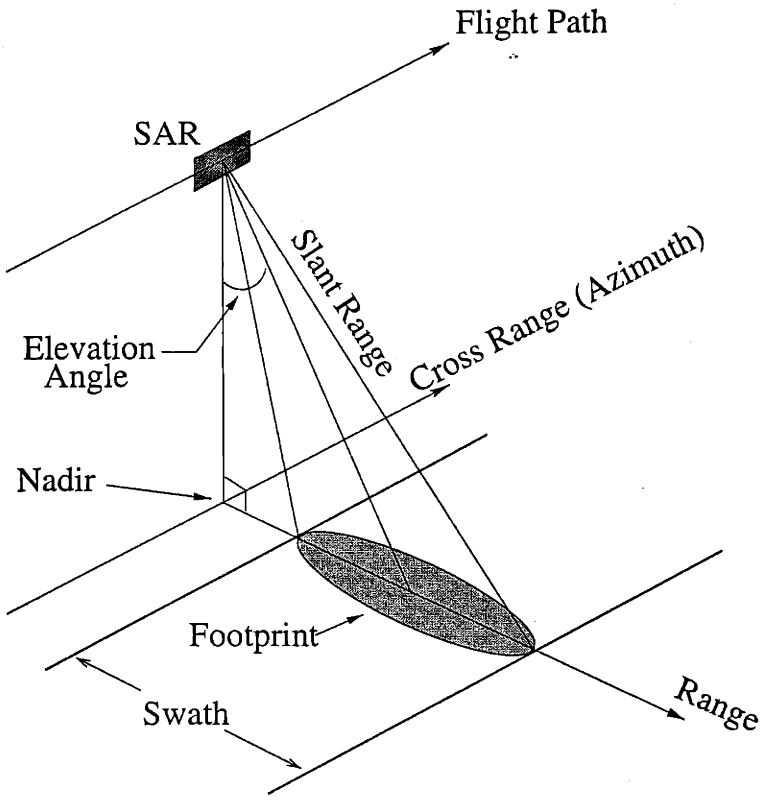


Figure 2.1: Illustration of terminology used in SAR imaging.

Letting x and y denote the range and azimuth/cross range coordinates respectively, each pixel in a SAR image approximates the complex valued *reflectiv-*

ity of the $dx \times dy$ spatial cell centered at (x, y). This reflectivity is written as $q(x, y) = m(x, y)e^{j\psi(x,y)}$ where $m(x, y)$ is the magnitude and $\psi(x, y)$ is the phase of the the reflection. Figure 2.2 illustrates this situation in which the reflection from (x, y) is being measured from one of many pulses sent at regularly spaced intervals of length Δ. The range of a cell is determined by the round–trip time delay of the radar pulse. An impulsive signal would thus be ideal for achieving high range resolution, but the use of such a waveform is impractical due to the difficulty in generating one for which the corresponding echo signal-to-noise ration (SNR) is large enough for reliable detection. To achieve high range resolution, microwave systems commonly use a technique called pulse compression in which a large–bandwidth signal, usually a linear frequency modulated (FM) chirp, is transmitted, and its return is processed to compress the duration of the pulse to achieve better resolution. The resulting range resolution is then the same as that from the compressed pulse. SAR achieves higher cross range resolution than conventional radar by taking advantage of the fact that the radar return from each scatterer in the antenna footprint will have a Doppler frequency shift. Given the range of the scatterer along with the speed and elevation of the SAR platform, the Doppler shift uniquely identifies the azimuthal location of the object. This shift manifests itself as a linear FM chirp in the cross range thus allowing the use of pulse compression to achieve high cross range resolution.

A linear FM pulse of bandwidth $\beta = T_p\alpha$ can be written as

$$s(t) = e^{j\pi\alpha t^2}\text{rect}\left(\frac{t}{T_p}\right) \tag{2.1}$$

where α is the chirp rate, T_p is the duration of the pulse, and

$$\text{rect}(t) = \begin{cases} 1 & \text{if } |t| \leq 1/2 \\ 0 & \text{otherwise.} \end{cases}$$

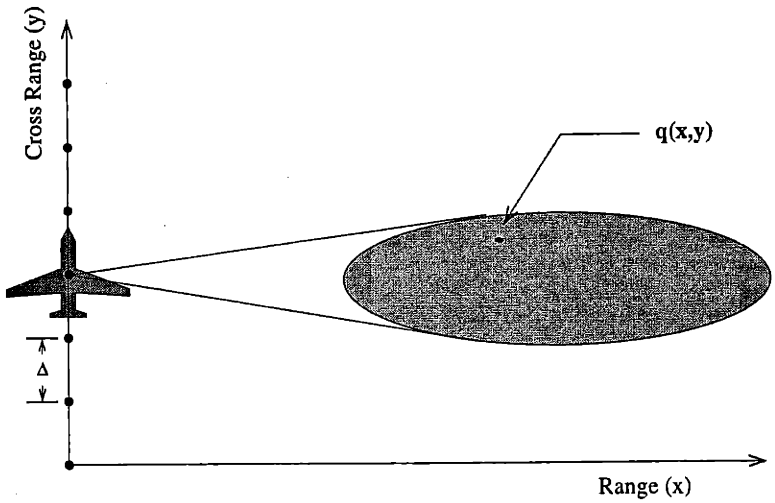


Figure 2.2: Geometry of cross range stripmap SAR imaging.

The transmitted signal can then be written as a carrier with frequency f_c modulating $s(t)$

$$\tilde{s}(t) = \mathrm{Re}\left\{s(t)\, e^{j2\pi f_c t}\right\}. \tag{2.2}$$

The return signal obtained by transmitting $\tilde{s}(t)$ is

$$\tilde{g}(t, n\Delta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(x, y-n\Delta)\, m(x,y)\, \mathrm{Re}\Bigg\{ s\left(t - \frac{2r_1(x, y-n\Delta)}{c}\right) e^{j\left[2\pi f_c\left(t-\frac{2r_1(x,y-n\Delta)}{c}\right)+\psi(x,y)\right]}\Bigg\} dx\, dy,$$

where c is the speed of light, $r_1(x,y)$ denotes the distance from the nadir to (x,y), and $w(x,y)$ represents the antenna gain pattern which is approximated as a separable function $w(x,y) = w_r(x)w_a(y)$ with $w_a(y)$ being even. The radar return $\tilde{g}(t, n\Delta)$ is then quadrature demodulated with the carrier frequency f_c as shown in Figure 2.3 to

produce

$$\bar{g}(t, n\Delta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w_r(x)\, w_a(n\Delta - y)\, q(x,y)\, \mathrm{e}^{j2kr_1(x,y-n\Delta)} \\ s\left(t - \frac{2r_1(x, y-n\Delta)}{c}\right) dx\, dy,$$

where $k = 2\pi f_c/c$ is the wave number.

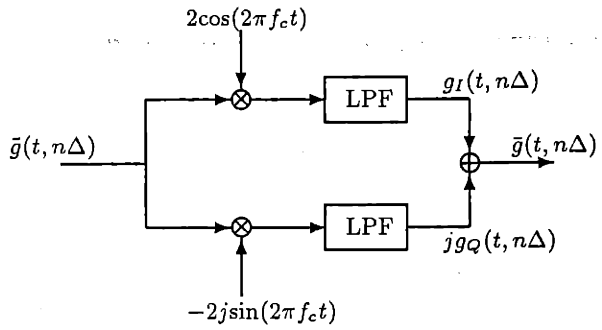


Figure 2.3: Quadrature demodulator decomposing the returned signal into in–phase and quadrature–phase components.

For airborne SAR, variations in y do not significantly affect $r_1(x, y - n\Delta)$ due to the limited extent of the footprint in the azimuthal direction. The range r_1 can then be approximated as x in $s(\cdot)$. Since the complex exponential is more sensitive to slight variations, a second order Taylor expansion is used to provide a more accurate approximation of r_1, or

$$r_1(x, y - n\Delta) = \sqrt{x^2 + (y - n\Delta)^2} \approx x + \frac{(y - n\Delta)^2}{2x}.$$

The approximated demodulated return is then rewritten as

$$\bar{g}(t, n\Delta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w_r(x)\, w_a(n\Delta - y)\, q(x, y)\, \mathrm{e}^{j2kx}\, \mathrm{e}^{jk(y-n\Delta)^2/x} \\ s\left(t - \frac{2x}{c}\right) dx\, dy.$$

With a change of variables ($\xi = x, \nu = y$, and $t = 2x/c$), the spatial form of the return can be written as

$$g(x, n\Delta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w_r(\xi)\, w_a(n\Delta - \nu)\, q(\xi, \nu)\, \mathrm{e}^{j2k\xi}\, \mathrm{e}^{j2k(\nu-n\Delta)^2/\xi} \\ s\left(\frac{2(x-\xi)}{c}\right) d\xi\, d\nu. \tag{2.3}$$

Because Eq. (2.3) has linear chirp-like terms for both the range and azimuth coordinates, pulse compression is used to estimate the reflectivity by applying a matched filter $h_r(x)h_a(x, n\Delta)$ with

$$h_r(x) = s^*\left(\frac{-2x}{c}\right)$$

and

$$h_a(x, n\Delta) = \mathrm{e}^{jk(n\Delta)^2/x}\mathrm{rect}\left(\frac{n\Delta}{L_s}\right),$$

where L_s defines the synthesized antenna length or equivalently the integration time. The resulting range resolution can then be shown to be $\delta r = c/2\beta$; by setting L_s equal to the antenna beamwidth; the azimuthal resolution can be derived as $\delta a = D/2$, where D is the diameter of the antenna [5].

2.2 Multiscale Framework

The framework introduced in [3] allows one to straightforwardly analyze processes that evolve in scale. In this section, we start by presenting the concept of a multiscale framework. We then proceed to detail the construction of a multiresolution sequence of real–valued images from a complex SAR image. These multiresolution images can be modeled using scale auto–regressive models to capture its evolution in scale. We then finish the section by relating how the image sequence and models can be reformulated in a multiscale representation.

2.2.1 Multiscale Representation

The framework developed in [3] and [6] provides a multiscale representation that allows efficient analysis of SAR data. The cornerstone of this framework is a *q-tree* which is a graph for which each vertex/node has one edge/branch connecting it to its parent node and "q" edges connecting it to its children nodes as illustrated in Figure 2.4. Denoting a node on the tree by s, the operator $\gamma_i(s)$ gives the i^{th} ancestor of s. Similarly, the operator $\alpha_i(s)$ is defined such that $\alpha_1(s), \alpha_2(s), \ldots, \alpha_q(s)$ denote the individual children of s. Processes evolving in scale can then be mapped to the nodes on the tree so that each level of the tree represents a particular scale realization of a process which proceeds from a coarse to a fine resolution as the tree is traversed from top to bottom. Multiscale processes have the particular property that they can be modeled as

$$\mathbf{x}(s) = A(s)\mathbf{x}(\gamma(s)) + B(s)\mathbf{w}(s) \tag{2.4}$$

for matrices A(s) and B(s) of appropriate size where $\mathbf{x}(s)$ represents a realization of the process at node s and $\mathbf{w}(s)$ is a white noise process. A fundamental property of multiscale processes is that they are Markov in scale. As a result, conditioned on the value at any node s, the nodes on the $q+1$ trees formed by removing node s are mutually independent as illustrated in Figure 2.5. This conditional independence affords highly efficient processing algorithms for processes that are multiscale in nature.

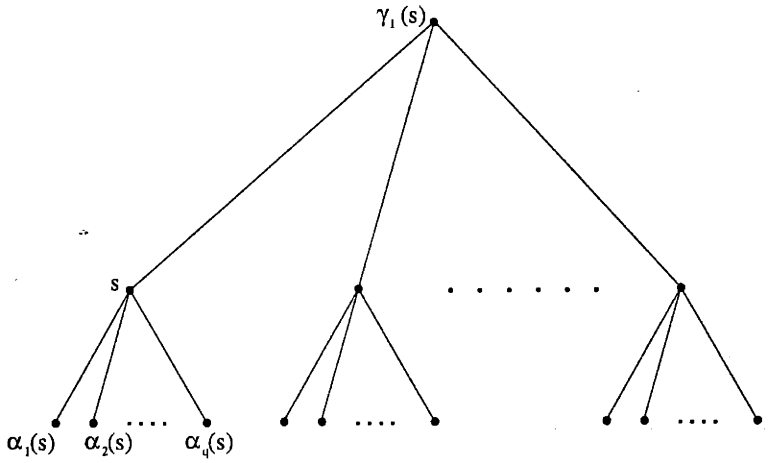


Figure 2.4: 3 levels of a q–tree for which each nonterminating node has q offspring.

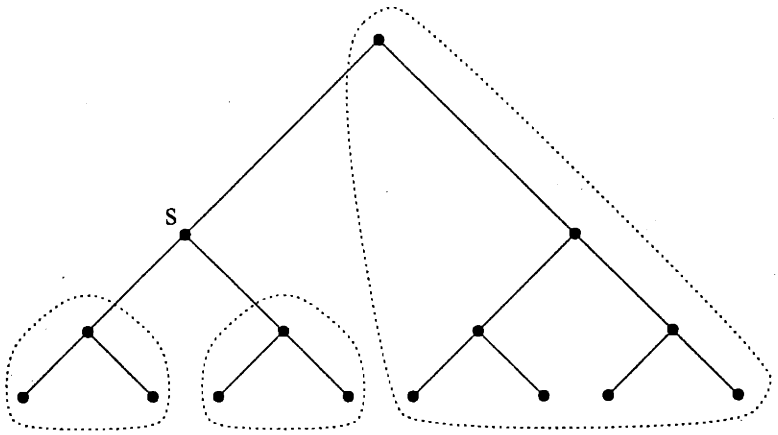


Figure 2.5: Illustration of the conditional independence for multiscale processes on a binary tree. Conditioned on the value at node s, the dotted boundaries show the 3 mutually independent processes.

2.2.2 Multiresolution SAR Image Formation

The natural starting point for finding a multiscale model for SAR imagery is to consider a multiresolution sequence of images. With access to the raw SAR echos, one could generate a sequence of images at arbitrary resolutions. In our case, however, we consider the situation in which one only has access to the finest–scale complex SAR image formed from the echos. We will use $\mathbf{Q}_1$ to denote this finest scale image and larger integer subscripts to denote correspondingly coarser resolutions. For any particular resolution, we use $[m, n]$ to denote the spatial coordinates for each dimension, where m and n are positive integers.

The multiresolution construction is initialized with the complex SAR image $\mathbf{Q}_1$ that conveys the reflectivity of the terrain. Coarser resolutions with a dyadically decreasing resolution are then obtained using the recursion

$$\mathbf{Q}_l[m, n] = \sum_{i=2m}^{2m+1} \sum_{j=2n}^{2n+1} \mathbf{Q}_{l-1}[i, j]. \tag{2.5}$$

The coarser resolution pixels are thus the coherent average of disjoint and adjacent 2×2 pixel blocks at the finer scale. The recursion in Eq. (2.5) is performed $L - 1$ times, where L represents the total number of desired resolutions. This particular multiresolution construction is motivated by the fact that each complex SAR pixel value represents the coherent average of the radar returns from objects within that pixel's resolution cell. The complex sum in Eq. (2.5) thus approximates the coherent average of the radar returns from objects within the coarser resolution cell corresponding to the total area spanned by the four finer resolution pixels. The set of multiresolution images $\{\mathbf{Q}_l\}_{l=1}^{L}$ may thus be viewed as an approximation of an image sequence generated by a SAR system programmed to operate at a dyadically varying resolution. The phase information present in the pixels, however, makes it impractical

to reliably model the scale evolution of the imagery. For many applications, such as target detection and recognition, it is sufficient to use the magnitudes of the reflectivities to produce statistically reliable information. A new set of images $\{\mathcal{I}_l\}_{l=1}^{L}$ is thus generated by taking the log-magnitude of the complex-valued nodes in $\{\mathbf{Q}_l\}_{l=1}^{L}$, i.e.

$$\mathcal{I}_l[m,n] = 20\log_{10}\left(\delta + |\mathbf{Q}_l[m,n]|\right),$$

where δ is a small positive real that constrains the argument of $\log_{10}$ to be a positive number. The resulting set of real–valued images can then be interpreted as a telescopic sequence of multiresolution images containing the magnitude of the radar reflectivities. These images can be naturally viewed as composing the layers of a *quadtree* which is a tree with $q = 4$. It can thus be visualized as a pyramid-shaped tree structure where each level corresponds to a specific resolution image as depicted in Figure 2.6. Note that although $\{\mathcal{I}_l\}_{l=1}^{L}$ compose a quadtree, it does *not* represent a multiscale process as no properties about its evolution in scale have been established.

2.2.3 Scale Auto-regressive Models

We first explicitly define a coarse scale operator ζ_i for the quadtree $\{\mathcal{I}_l\}_{l=1}^{L}$ as $\zeta_i[m,n] = [\lceil m/2^i \rceil, \lceil n/2^i \rceil]$ where $\lceil\cdot\rceil$ represents the greatest integer function. Using ζ_i, scale auto–regressive (AR) models can be written in the form

$$\mathcal{I}_l[m,n] = \varepsilon_l[m,n] + \sum_{i=1}^{R} a_{l,i}\mathcal{I}_{l+i}\left(\zeta_i[m,n]\right),$$

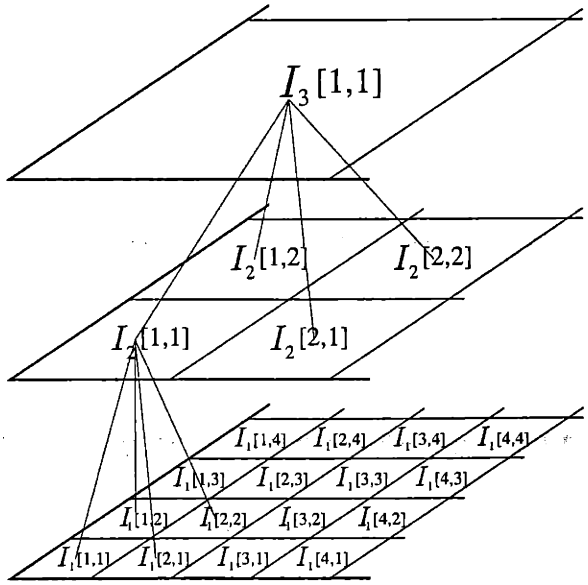


Figure 2.6: 3 levels of a quadtree where each node at the coarser scales has four offspring.

where R is the order of the regression, and $\epsilon_l[m,n]$ is a white noise process which can be viewed as the modeling error. In order to have an unbiased prediction error, we decompose $\epsilon_l[m,n]$ into a white zero mean process $\tilde{\epsilon}_l[m,n]$ and a constant α_l to yield

$$\mathcal{I}_l[m,n] = \alpha_l + \tilde{\epsilon}_l[m,n] + \sum_{i=1}^{R} a_{l,i}\mathcal{I}_{l+i}\left(\zeta_i[m,n]\right). \tag{2.6}$$

The set of AR coefficients $\{a_{l,i}\}_{i=1}^{R}$ for level l are independent of $[m,n]$ and are thus constant for a fixed value of l as the notation suggests. The $\{a_{l,i}\}$ then give the linear dependencies between the various scales of the representation; the $\{\alpha_l\}$ give the linear dependencies of the various scales on the space of constants. The scale AR model in Eq. (2.6), in essence, statistically captures the evolution in scale on the quadtree $\{\mathcal{I}_l\}_{l=1}^{L}$. Because we use multiple models of this form to characterize different terrain types, Eq. (2.6) is modified to reflect this additional degree of freedom to result in a

model of the form

$$\mathcal{I}_l[m,n] = \alpha_{l,\mu} + \tilde{\varepsilon}_{l,\mu}[m,n] + \sum_{i=1}^{R} a_{l,i,\mu}\mathcal{I}_{l+i}\left(\zeta_i[m,n]\right), \tag{2.7}$$

where μ denotes the classification of $\mathcal{I}_l[m,n]$.

For a given classification, the parameters in Eq. (2.7) are determined by minimizing the energy in $\tilde{\varepsilon}_{l,\mu}[m,n]$ over a homogeneous training image of the corresponding class. Specifically, the coefficients are selected according to the least squares criterion

$$\begin{aligned}\hat{\mathbf{a}}_{l,\mu} = \arg\Bigg\{ \min_{\mathbf{a}_{l,\mu}} \Bigg\{ \sum_{m,n} \Big(\mathcal{I}_l[m,n] - a_{l,1,\mu}\mathcal{I}_{l+1}\left(\zeta_1[m,n]\right) - a_{l,2,\mu}\mathcal{I}_{l+2}\left(\zeta_2[m,n]\right) \\ - \cdots - a_{l,R,\mu}\mathcal{I}_{l+R}\left(\zeta_R[m,n]\right) - \alpha_{l,\mu} \Big)^2 \Bigg\} \Bigg\},\end{aligned} \tag{2.8}$$

where $\mathbf{a}_{l,\mu} = [a_{l,1,\mu};\ a_{l,2,\mu};\ \ldots;\ a_{l,R,\mu};\ \alpha_{l,\mu}]$ is a column vector. This minimization is performed for each value of $l \in \{1, 2, \ldots, L-1\}$ thus giving the complete model $\{\hat{\mathbf{a}}_{l,\mu}\}_{l=1}^{L-1}$ for classification μ. In this framework, the quadtree $\{\mathcal{I}_l\}_{l=1}^{L}$ then provides a structure suitable for adapted modeling thus affording efficient segmentation and compression.

The quadtree $\{\mathcal{I}_l\}_{l=1}^{L}$ with Eq. (2.7) clearly does not represent a multiscale process except for the case of $R = 1$, but via state augmentation, it can be reformulated in the form of Eq. (2.4). Specifically, we define node s to be the 3–tuple (l, m, n) and define the associated state vector as

$$\mathbf{x}(s) = \mathbf{x}_l[m,n] = \Big[\mathcal{I}_l[m,n], \mathcal{I}_{l+1}(\zeta_1[m,n]), \ldots, \mathcal{I}_{l+R-1}(\zeta_{R-1}[m,n]), 1\Big]^T. \tag{2.9}$$

The coarse scale operator γ_i is then given by

$$\gamma_i(s) = \gamma_i(l, m, n) = \left(l + i, \left\lceil \frac{m}{2^i} \right\rceil, \left\lceil \frac{n}{2^i} \right\rceil\right). \tag{2.10}$$

Eq. (2.8) can then be written as

$$\mathbf{x}(s) = \begin{bmatrix} a_{l,1,\mu} & a_{l,2,\mu} & \dots & a_{l,R-1,\mu} & a_{l,R,\mu} & \alpha_{l,\mu} \\ 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix} \mathbf{x}(\gamma_1(s)) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tilde{\varepsilon}(s), \tag{2.11}$$

which makes it explicit that $\mathbf{x}$ constitutes a multiscale process that can be mapped to a vector valued quadtree. It should be noted that the first row of A in Eq. (2.11) is equal to $\hat{\mathbf{a}}_{l,\mu}^T$. The modeling matrices are thus constant for a given scale and model classification. This fact affords us great utility in SAR image segmentation and compression later in this thesis.

2.3 Hypothesis Testing

In this section, we discuss the application of a hypothesis test to determine which hypothesis about an underlying source is true given its observation. Our objective is thus: given an N–dimensional observation vector $\mathbf{y}$, partition the observation space $\mathbb{R}^N$ into M convex regions $\{Z_0, Z_1, \dots, Z_{M-1}\}$ corresponding to the M hypotheses $\{H_0, H_1, \dots, H_{M-1}\}$ so that $\mathbf{y} \in Z_i$ results in the decision that $\mathbf{y}$ was generated under hypothesis H_i. The particular decision rule will depend not only on the dependencies

between the H_i and $\mathbf{y}$ given by the likelihoods $p_{\mathbf{Y}|H}(\mathbf{y}|H_i)$, but also on the apriori probabilities $P_i \triangleq Pr[H = H_i]$ and the cost function C_{ij} representing the cost of deciding H_i is true when H_j is true.

We consider here the binary hypothesis case which can be extended to the M–ary case through M(M-1)/2 binary tests. The performance of a given decision rule is measured by its average cost (Bayes risk) conditioned on the observation, i.e.

$$E[\text{cost of } \hat{H} = H_i|\mathbf{y}] = \sum_{i=0}^{1} P_{H|\mathbf{Y}}(H_j|\mathbf{y})\, C_{ij}. \tag{2.12}$$

The optimum Bayes risk decision rule is the one that minimizes this cost and is thus

$$C_{00}P_{H|\mathbf{Y}}(H_0|\mathbf{y}) + C_{01}P_{H|\mathbf{Y}}(H_1|\mathbf{y}) \underset{\hat{H}=H_0}{\overset{\hat{H}=H_1}{\gtrless}} C_{10}P_{H|\mathbf{Y}}(H_0|\mathbf{y}) + C_{11}P_{H|\mathbf{Y}}(H_1|\mathbf{y}).$$

Rearranging terms to group underlying hypotheses together, we get

$$(C_{01} - C_{11})\, P_{H|\mathbf{Y}}(H_1|\mathbf{y}) \underset{\hat{H}=H_0}{\overset{\hat{H}=H_1}{\gtrless}} (C_{10} - C_{00})\, P_{H|\mathbf{Y}}(H_0|\mathbf{y}).$$

By grouping the posterior probabilities together, we then obtain

$$\frac{P_{H|\mathbf{Y}}(H_1|\mathbf{y})}{P_{H|\mathbf{Y}}(H_0|\mathbf{y})} \underset{\hat{H}=H_0}{\overset{\hat{H}=H_1}{\gtrless}} \frac{(C_{10} - C_{00})}{(C_{01} - C_{11})}.$$

By applying Bayes Rule, the binary decision rule can then be written as a likelihood

ratio test (LRT)

$$L(\mathbf{y}) \triangleq \frac{p_{\mathbf{Y}|H}(\mathbf{y}|H_1)}{p_{\mathbf{Y}|H}(\mathbf{y}|H_0)} \underset{\hat{H}=H_0}{\overset{\hat{H}=H_1}{\gtrless}} \frac{(C_{10}-C_{00})P_0}{(C_{01}-C_{11})P_1} \triangleq \eta. \tag{2.13}$$

A special case of the LRT is the Maximum Likelihood (ML) detector which uses equal apriori probabilities (i.e. $P_0 = P_1$) and $C_{ij} = \delta_{ij}$ where δ_{ij} is the Kronecker delta function, thereby resulting in the threshold η being equal to 1. The ML detector has a simple and computationally efficient form if the likelihoods $p_{\mathbf{Y}|H}(\mathbf{y}|H_0)$ and $p_{\mathbf{Y}|H}(\mathbf{y}|H_1)$ are N-variate Gaussian densities with means $\mathbf{m}_0$ and $\mathbf{m}_1$ and covariances Λ_0 and Λ_1, respectively. By incorporating the normalizing coefficients of the likelihoods into the right-hand side of the inequality and taking the logarithm of both sides of Eq. (2.13), the log-likelihood ratio test is obtained as

$$\ell(\mathbf{y}) \triangleq (\mathbf{y}-\mathbf{m}_0)^T\Lambda_0^{-1}(\mathbf{y}-\mathbf{m}_0) - (\mathbf{y}-\mathbf{m}_1)^T\Lambda_1^{-1}(\mathbf{y}-\mathbf{m}_1) \underset{\hat{H}=H_0}{\overset{\hat{H}=H_1}{\gtrless}} \log\frac{|\Lambda_1|}{|\Lambda_0|} \triangleq \lambda. \tag{2.14}$$

The quantity $\ell(\mathbf{y})$ is commonly referred to as the *sufficient statistic* because it conveys all the relevant information in $\mathbf{y}$ as a simple scalar. The optimal decision rule is then a simple comparison of $\ell(\mathbf{y})$ to the constant λ.

2.4 Entropy and Lossless Data Compression

The entropy of a discrete random variable X taking on values from the set $\mathcal{X}$ is a quantitative measure of the unpredictability in realizations of X. For a discrete

random variable X with probability mass function $p_X(x)$, its Shannon entropy is defined as

$$H_K(X) = \sum_{x \in \mathcal{X}} -p_X(x) \log_K(p_X(x)), \tag{2.15}$$

where K is the size of the set $\mathcal{X}$. As we shall specifically focus on the Shannon entropy, all references to entropy in this section will imply the Shannon entropy. For a given set $\mathcal{X}$, the distribution over $\mathcal{X}$ that maximizes the entropy is the uniform distribution. This is consistent with intuition since given a fixed set size K, a uniform distribution on $\mathcal{X}$ would produce the least predictable realizations. Another consequence of Eq. (2.15) is that if we have a set of random variables $\{X_1, X_2, \ldots, X_N\}$, then the N–tuple $(X_1, X_2, \ldots, X_N)$ is itself a random variable with entropy given by Eq. (2.15) using the joint distribution, or equivalently,

$$H_K(X_1, X_2, \ldots, X_N) = H_K(X_1) + \sum_{n=2}^{N} H_K(X_n | X_1, \ldots, X_{n-1}). \tag{2.16}$$

If the X_n are independent and identically distributed (IID), then Eq. (2.16) simplifies to

$$H_K(X_1, X_2, \ldots, X_N) = N H_K(X_1). \tag{2.17}$$

The entropy rate of a process $\mathbf{X} = (X_n)_{n \in \mathbb{Z}_+}$ is defined as the limit, if it exists, of $H_K(X_1, \ldots, X_N)$ as $N \to \infty$, i.e.

$$\mathcal{H}_K(\mathbf{X}) \triangleq \lim_{N \to \infty} \frac{1}{N} H_K(X_1, X_2, \ldots, X_N). \tag{2.18}$$

A *source code* $\mathcal{C}$ for X is defined as a mapping from $\mathcal{X}$ to $\mathcal{D}$ where $\mathcal{D}$ is the set of finite length strings of symbols from a D–ary alphabet which without loss of

generality we take to be $\{0, 1, \ldots, D-1\}$. The codeword corresponding to x will be denoted by $\mathcal{C}(x)$ and the length of $\mathcal{C}(x)$ will be denoted by $l(x)$. A source code is called *uniquely decodable*, if for any two sequences $\mathbf{X}_a$ and $\mathbf{X}_b$ of elements in $\mathcal{X}$, the coded sequences $(C(x_{a,i}))_{i\in\mathbb{Z}}$ and $(C(x_{b,i}))_{i\in\mathbb{Z}}$ are distinct. In this section, our attention will be focused on such codes. The compression[1] efficiency of a code is then given by its *expected length*

$$L(\mathcal{C}) = \sum_{x\in\mathcal{X}} p_X(x) l(x). \tag{2.19}$$

The objective of data compression is then to find a mapping $\mathcal{C} : \mathcal{X} \to \mathcal{D}$ that achieves the minimum expected length, i.e. find

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \{L(\mathcal{C})\}. \tag{2.20}$$

With the use of Lagrange multipliers it can be shown that $l^*(x_i) = -\log_D(p_X(x_i))$ gives the minimum value of the $L(\mathcal{C})$ which is $H_D(X)$ [7]. Because these $l^*(x_i)$ may take on nonintegral values, this only represents a lower bound on the expected length. But, the expected length of a code for which $l(x_i) = \lceil l^*(x_i) \rceil$ is less than $H_D(X) + 1$ and thus

$$H_D(X) \leq L(\mathcal{C}) < H_D(X) + 1. \tag{2.21}$$

If the X_i are IID, then a group of N symbols $(X_1, \ldots, X_N)$ can be thought of as a supersymbol from the set $\mathcal{X}^N$ which has expected length

$$H_D(X_1, X_2, \ldots, X_N) \leq E[l(X_1, X_2, \ldots, X_N)] < H_D(X_1, X_2, \ldots, X_N) + 1.$$

[1]Note that all references to compression in this section refer to lossless compression.

By applying Eq. (2.17) and simplifying we get

$$NH_D(X_i) \leq NL_N < NH_D(X_i) + 1$$
$$H_D(X_i) \leq L_N < H_D(X_i) + \frac{1}{N},$$

where L_N is the expected codeword length per input symbol. We thus see that by grouping symbols together, we can construct a code whose expected symbol length is arbitrarily close to the entropy of X, and if $\mathbf{X}$ is a stationary process, then $L_N \rightarrow \mathcal{H}_D(\mathbf{X})$ as $N \rightarrow \infty$.

2.4.1 Arithmetic Coding

We discuss here a particularly popular form of data compression called arithmetic coding. This compression algorithm has many attractive features including

1. universal encoding capabilities, i.e. the ability to adapt to the statistics of arbitrary sources,
2. the existence of computationally and memory efficient implementations, and
3. the ability to code, on the average, a sequence within two bits of its entropy rate.

The idea behind arithmetic coding is that a sequence of symbols can be represented as the approximation of a real number in the half open interval $[0, 1) = \{y \in \mathbb{R} : 0 \leq y < 1\}$ based on estimated symbol probabilities. For a sequence of length N originating from a source producing symbols from the set $\mathcal{X} = \{0, \ldots, K-1\}$, any

number in the interval $[a_N, b_N)$ can be used to represent the entire sequence where a_N and b_N are given by the recursion

$$a_1 = 0, \tag{2.22a}$$

$$b_1 = 1, \tag{2.22b}$$

$$a_n = a_{n-1} + \sum_{l=1}^{x_n - 1} p_X(x_l), \tag{2.22c}$$

$$b_n = a_n + (b_{n-1} - a_{n-1}) p_X(x_n) \tag{2.22d}$$

where x_n is the symbol corresponding to the n^{th} element in the sequence. So for arithmetic coding, the code C actually maps $2^{\mathcal{X}}$ into $2^{\mathcal{D}}$, where $2^{\mathcal{X}}$ represents the power set of $\mathcal{X}$ defined as $\bigcup_{n \in \mathbb{N}} \mathcal{X}^n$. This procedure is illustrated by an example in Figure 2.7 for a source that has three symbols $\{0, 1, 2\}$. After the entire sequence is processed and

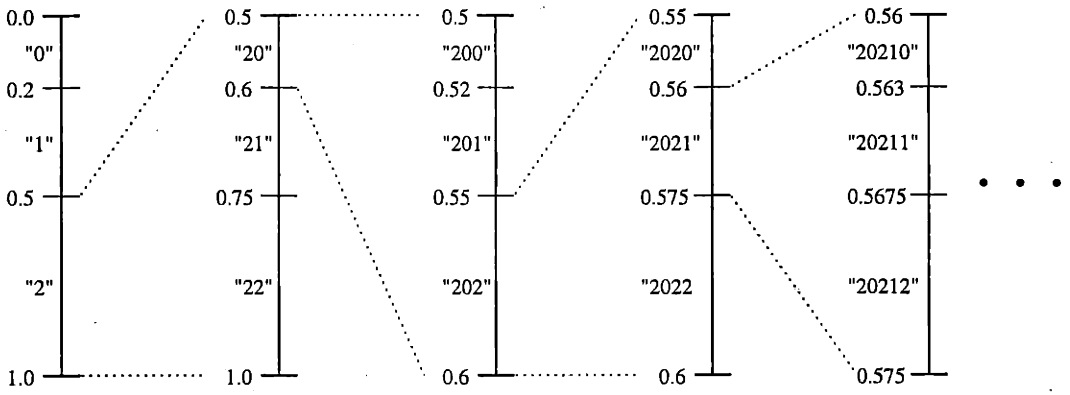


Figure 2.7: Example of infinite precision arithmetic coding for a sequence starting with "2021" and symbol probabilities: p(0)=0.2, p(1)=0.3, and p(2)=0.5.

a_N and b_N have been determined, the source sequence can be coded as any real number in the interval $[a_N, b_N)$. The number with the shortest representation, e.g. binary, is the one encoded. It is now clear why the sub-intervals are partitioned proportionally to the symbol probabilities. A larger final interval width gives a larger range of numbers from which to choose, and hence increasing the probability of being able to choose a smaller coded sequence. A problem however arises if the arithmetic encoder

is implemented using floating point arithmetic. When coding arbitrarily long source sequences, one may encounter round-off inconsistencies if the encoding and decoding are carried out with different computer architectures due to the assumption that the operations in the Eq. (2.22) can be carried out with infinite precision. These issues have been solved by the use of finite precision integer arithmetic and are discussed in [8] and [9]. Since the algorithm can be adapted without changing the underlying principle, they will not be discussed further.

The performance of the arithmetic coder, or any entropy coder, is highly dependent on its ability to accurately estimate the symbol probabilities in order to optimally partition the intervals in the recursion. The coded file could even be larger than the original if the estimated probability distribution significantly deviates from the underlying distribution. There is thus a strong motivation to accurately model the symbol probability distribution. Unlike the example in Figure 2.7, the estimated symbol probabilities do not have to be constant. In fact, it is straightforward to extend the arithmetic coder to use an adaptive distribution. One can easily incorporate an addition step in the iteration in Eq. (2.22) that uses the relative frequency of the past symbols to estimate the probabilities for the next symbol. Such a method is attractive since for a stationary source, the estimated symbol distribution in the arithmetic coder will converge to the actual symbol distribution under an arbitrary initial estimate.

2.5 Wavelets

In recent years, wavelets have found use in many applications of science and engineering. They present a computationally efficient multiresolution signal decomposition

that corresponds to projections onto different portions of the frequency spectrum. The processing in each subspace can then be adapted to take advantage of particular signal characteristics in those bands. In Section 2.5.1, we briefly discuss how wavelets arise out of the study of a multiresolution analysis of square integrable functions, and in Section 2.5.2, we describe how octave band filter banks can be used to efficiently implement discrete time wavelet series expansions of square summable sequences. In Section 2.5.3, we discuss a particular extension to two–dimensional sequences using separable functions to generate the basis. We then conclude with Section 2.5.4 by generalizing the family of orthonormal bases one can construct from wavelets.

2.5.1 Continuous Time Wavelets

In this section, we consider how wavelets arise from multiresolution representations of continuous time functions. We consider here the Hilbert space $L^2(\mathbb{R})$ of real–valued square integrable functions whose domain is the real line with the inner product defined as

$$\langle x(\cdot), y(\cdot) \rangle = \int_{-\infty}^{+\infty} x(t)y(t)dt, \tag{2.23}$$

for $x(\cdot), y(\cdot) \in L^2(\mathbb{R})$. To obtain a time–scale representation of functions in $L^2(\mathbb{R})$, we define a family of real valued functions as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \tag{2.24}$$

for $a > 0$ and $b \in \mathbb{R}$ where $\psi(\cdot)$ is a fixed function, called the *mother wavelet*, that is well localized in both time and frequency [10]. The factor $1/\sqrt{a}$ ensures that $\|\psi_{a,b}(\cdot)\| = \|\psi(\cdot)\|$, i.e. that each function in the family has the same norm, which

without loss of generality will be taken to be one. We can then define the continuous wavelet transform (CWT) of $x(\cdot)$ as

$$\mathcal{W}_x(a,b) = \langle \psi_{a,b}(\cdot), x(\cdot) \rangle = \int_{-\infty}^{+\infty} \psi_{a,b}(t) x(t) dt. \tag{2.25}$$

An inverse transform exists if $\psi(\cdot)$ satisfies

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{|\omega|^2}\, d\omega < +\infty,$$

where $\Psi(\omega) = \int_{-\infty}^{+\infty} \psi(t) \mathrm{e}^{-j\omega t}\, dt$ is the Fourier transform of $\psi(t)$. For such $\psi(\cdot)$, $x(\cdot)$ can be reconstructed by

$$x(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{a^2} \mathcal{W}_x(a,b) \psi_{a,b}(t)\, db\, da. \tag{2.26}$$

In practical applications, the CWT can only be computed on a discrete set of points $(a_k, b_k)_{k \in \mathbb{Z}}$. A particular choice of $\{(a_k, b_k)\}$ are those that give the subset of functions in Eq. (2.24) which have the form

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k), \tag{2.27}$$

where $(j,k) \in \mathbb{Z}^2$. It is then possible to choose $\psi(\cdot)$ so that this family of functions forms a basis satisfying the L^2 stability criterion that there exist two constants c and C such that

$$c \sum_j \sum_k |a_{j,k}|^2 \leq \int \Big| \sum_j \sum_k a_{j,k} \psi_{j,k}(t) \Big|^2 dt \leq C \sum_j \sum_k |a_{j,k}|^2, \tag{2.28}$$

for any choice of coefficients $a_{j,k}$.

We next introduce the idea of a *multiresolution analysis* which is a sequence

of approximation subspaces $\{V_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$ that satisfies:

1. the V_j are generated by a *scaling function* $\varphi(\cdot) \in L^2(\mathbb{R})$ in the sense that for each fixed j, the family of functions

$$\{\varphi_{j,k}(\cdot)|\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k)\}_{k\in\mathbb{Z}}$$

spans the space V_j and satisfies the L^2 stability condition in Eq. (2.28),

2. the spaces are embedded, i.e. $V_j \subset V_{j+1}$ for all j, and

3. the orthogonal projectors P_j onto V_j satisfy

$$\lim_{j\to+\infty} P_j x(\cdot) = x(\cdot) \qquad \text{and} \qquad \lim_{j\to-\infty} P_j x(\cdot) = 0, \textit{and}$$

for any $x(\cdot) \in L^2(\mathbb{R})$.

Several consequences of this definition are that:

1. $x(\cdot) \in V_j$ if and only if $x(2\cdot) \in V_{j+1}$,

2. V_j is invariant under translations of 2^{-j}, and

3. $\varphi(\cdot)$ is the solution of a two–scale equation

$$\varphi(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} h[n]\varphi(2t - n). \tag{2.29}$$

If we define $A_j x(\cdot)$ as the reconstruction of $x(\cdot)$ from its projections onto V_j, then

when integer translates of $\varphi(\cdot)$ are orthonormal,

$$A_j x(\cdot) = \sum_{k\in\mathbb{Z}} \langle \varphi_{j,k}(\cdot), x(\cdot)\rangle \varphi_{j,k}(\cdot). \tag{2.30}$$

If integer translates of $\varphi(\cdot)$ are not orthonormal, then we can construct a dual scaling function $\tilde{\varphi}(\cdot)$ such that

$$\langle \tilde{\varphi}(\cdot), \varphi(\cdot - k)\rangle = \delta_{k,0} \qquad \text{and} \tag{2.31}$$

$$A_j x(\cdot) = \sum_{k\in\mathbb{Z}} \langle \tilde{\varphi}_{j,k}(\cdot), x(\cdot)\rangle \varphi_{j,k}(\cdot), \tag{2.32}$$

where $\tilde{\varphi}(\cdot)$ is the solution to a two–scale equation

$$\tilde{\varphi}(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} \tilde{h}[n]\tilde{\varphi}(2t - n). \tag{2.33}$$

In order for the $\varphi(\cdot)$ to be orthonormal, $h[\cdot]$ must satisfy

$$\sum_{k\in\mathbb{Z}} h^*[n]h[n + 2k] = \delta_{k,0}. \tag{2.34}$$

A necessary condition for the duality of $\varphi(\cdot)$ and $\tilde{\varphi}(\cdot)$ is that

$$\sum_{k\in\mathbb{Z}} h^*[n]\tilde{h}[n + 2k] = \delta_{k,0}. \tag{2.35}$$

Once the scaling function has been determined, in order to find its associated wavelet, we define

$$g[n] = (-1)^n \tilde{h}[1 - n], \qquad \text{and} \qquad \tilde{g}[n] = (-1)^n h[1 - n]. \tag{2.36}$$

The wavelet and its dual can then be defined as

$$\psi(t) = \sqrt{2}\sum_{k\in\mathbb{Z}} g[n]\varphi(2t-k) \qquad \text{and} \tag{2.37a}$$

$$\tilde{\psi}(t) = \sqrt{2}\sum_{k\in\mathbb{Z}} \tilde{g}[n]\tilde{\varphi}(2t-k). \tag{2.37b}$$

It then follows that

$$A_{j+1}x(\cdot) - A_j x(\cdot) = \sum_{k\in\mathbb{Z}} \langle \tilde{\psi}_{j,k}(\cdot), x(\cdot)\rangle \psi_{j,k}(\cdot) \tag{2.38}$$

thus providing a multiscale decomposition. We also have

$$\langle \tilde{\psi}_{j',k'}(\cdot), \psi_{j,k}(\cdot)\rangle = \delta_{j,j'}\delta_{k,k'}$$

in the dual case. Note that in the orthonormal case, each item is its own dual so that $\varphi(\cdot) = \tilde{\varphi}(\cdot)$, $\psi(\cdot) = \tilde{\psi}(\cdot)$, $h[\cdot] = \tilde{h}[\cdot]$, and $g[\cdot] = \tilde{g}[\cdot]$.

2.5.2 Discrete Time Wavelet Series

We describe here an efficient method to compute the discrete time wavelet series (DTWS) expansion of a signal in the space of square summable sequences $\ell^2(\mathbb{Z})$ on the field of real numbers with inner product

$$\langle x[\cdot], y[\cdot]\rangle = \sum_{l\in\mathbb{Z}} x[l]y[l], \tag{2.39}$$

for $x[\cdot], y[\cdot] \in \ell^2(\mathbb{Z})$. We start with the assumption that the multiresolution axioms hold and consider a function $x(\cdot) \in V_j$ which can be written as

$$x(t) = \sum_{n\in\mathbb{Z}} X^{(j)}[n]\varphi(2t-n) \tag{2.40}$$

where $X^{(j)}[n] = \langle \tilde{\varphi}(2^j \cdot -n), x(\cdot)\rangle$. We denote by W_{j-1} the orthogonal complement of V_{j-1} in V_j, i.e. $V_j = V_{j-1} \oplus W_{j-1}$. Eqs. (2.29) and (2.37) then yield

$$X^{(j-1)}[n] = \sum_{k\in\mathbb{Z}} \tilde{h}[k-2n]X^{(j)}[k] \qquad \text{and} \tag{2.41a}$$

$$Y^{(j-1)}[n] = \sum_{k\in\mathbb{Z}} \tilde{g}[k-2n]X^{(j)}[k]. \tag{2.41b}$$

where $X^{(j-1)}[n]$ and $Y^{(j-1)}[n]$ denote the projections of $x(\cdot)$ onto V_{j-1} and W_{j-1}, respectively. We see from Eq. (2.38) that $Y^{(j-1)}[\cdot]$ conveys the details $A_j x(\cdot) - A_{j-1}x(\cdot)$ lost by taking the projection onto the approximation space V_{j-1}. Eq. (2.41) is also equivalent to filtering $X^{(j)}[\cdot]$ with $\tilde{h}[-\cdot]$ and $\tilde{g}[-\cdot]$ and downsampling the result by two. Similarly, we can apply Eq. (2.41) to $X^{(j-1)}[\cdot]$ in place of $X^{(j)}[\cdot]$ to obtain the projections onto V_{j-2} and W_{j-2}. By iterating, this procedure J times, we obtain a filter bank with J stages as shown in Figure 2.8(a). We will let $h^{(j)}[\cdot]$ denote the impulse response equivalent to j iterations of the low–pass channel and $g^{(j)}[\cdot]$ denote the impulse response equivalent to going through the first $j-1$ iterations of the low–pass channel followed by the j^{th} high–pass channel. The DTWS expansion of $x[\cdot] \in \ell^2(\mathbb{Z})$ over J octaves and a dyadic sampling grid can now be defined as

$$x[n] = \sum_{j=1}^{J}\sum_{k\in\mathbb{Z}} Y^{(j)}[k]\tilde{g}^{(j)}[n-2^j k] + \sum_{k\in\mathbb{Z}} X^{(J)}[k]\tilde{h}^{(J)}[n-2^J k] \tag{2.42}$$

where

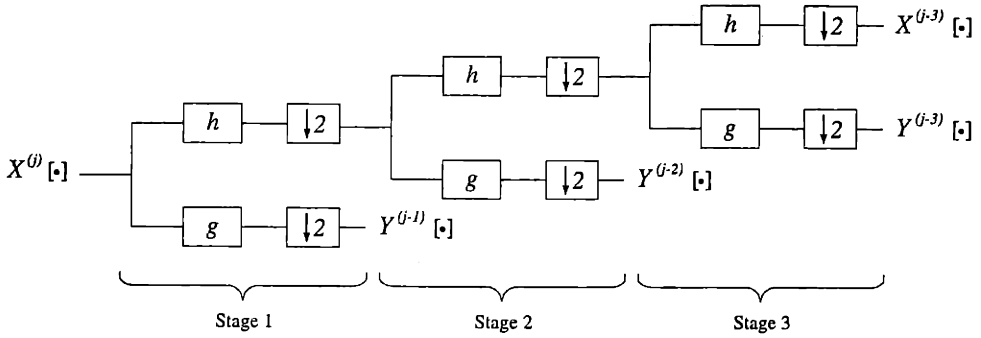
$$Y^{(j)}[2k+1] = \langle g^{(j)}[2^j k - \cdot], x[\cdot]\rangle = \langle \tilde{g}^{(j)}[\cdot - 2^j k], x[\cdot]\rangle \qquad \text{and} \tag{2.43a}$$

$$X^{(J)}[2k] = \langle h^{(J)}[2^J k - \cdot], x[\cdot]\rangle = \langle \tilde{h}^{(J)}[\cdot - 2^J k], x[\cdot]\rangle. \tag{2.43b}$$

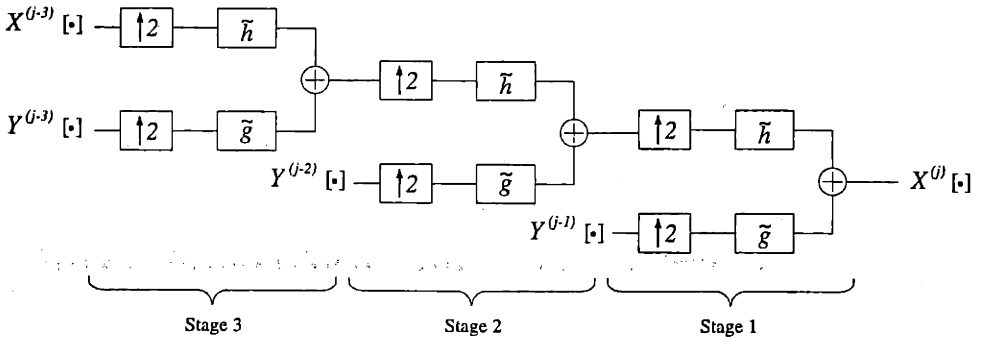
This DTWS provides a series expansion in the orthonormal basis

$$\{\tilde{g}^{(j)}[2^j k - \cdot], \tilde{h}^{(J)}[2^J k - \cdot]\}_{k\in\mathbb{Z}, j\in\{1,2,\ldots,J\}}$$

of $\ell^2(\mathbb{Z})$. Furthermore, the DTWS decomposition and reconstruction of a sequence can be carried out as illustrated in Figure 2.8 using efficient discrete–time signal processing operations.



(a) Projection of $f^{(j)}[\cdot]$ onto approximation and detail subspaces.



(b) Reconstruction of $f^{(j)}[\cdot]$ from projections.

Figure 2.8: Illustration of a 3 stage tree structured filter bank giving the DTWS expansion of $X^{(j)}[\cdot]$.

2.5.3 Separable Two-dimensional Wavelets

Extending the multiresolution analysis beyond one dimensional (1–D) signals increases the number of degrees of freedom in choosing a scaling function and its associated wavelet. For simplicity of construction, we focus our attention to 2–D bases obtained by taking Kronecker products of a 1–D scaling function and its associated wavelet thus generating separable functions as the basis for $L^2(\mathbb{R}^2)$. The 2–D scaling function and its three wavelets can then be written as

$$\varphi_{j,m,n}(t_1,t_2) = \varphi_{j,m}(t_1)\varphi_{j,n}(t_2) = 2^j\varphi(2^jt_1-m)\varphi(2^jt_2-n), \tag{2.44a}$$

$$\psi^{(1)}_{j,m,n}(t_1,t_2) = \varphi_{j,m}(t_1)\psi_{j,n}(t_2) = 2^j\varphi(2^jt_1-m)\psi(2^jt_2-n), \tag{2.44b}$$

$$\psi^{(2)}_{j,m,n}(t_1,t_2) = \psi_{j,m}(t_1)\varphi_{j,n}(t_2) = 2^j\psi(2^jt_1-m)\varphi(2^jt_2-n), \text{ and} \tag{2.44c}$$

$$\psi^{(3)}_{j,m,n}(t_1,t_2) = \psi_{j,m}(t_1)\psi_{j,n}(t_2) = 2^j\psi(2^jt_1-m)\psi(2^jt_2-n). \tag{2.44d}$$

The separability of the scaling function and wavelets allows the decomposition of a signal to be efficiently carried out with two–channel filter banks as for the 1–D signal case. The decomposition of a signal $x(\cdot,\cdot\cdot) \in V_j$ onto the approximation space

$$V_{j-1} = \text{Span}(\{\varphi_{j-1,m,n}(\cdot,\cdot\cdot)\}_{m,n})$$

and the detail spaces

$$W^{(1)}_{j-1} = \text{Span}(\{\psi^{(1)}_{j-1,m,n}(\cdot,\cdot\cdot)\}_{m,n}),$$
$$W^{(2)}_{j-1} = \text{Span}(\{\psi^{(2)}_{j-1,m,n}(\cdot,\cdot\cdot)\}_{m,n}), \text{ and}$$
$$W^{(3)}_{j-1} = \text{Span}(\{\psi^{(3)}_{j-1,m,n}(\cdot,\cdot\cdot)\}_{m,n}).$$

can be accomplished with two stages of the DTWS decomposition. The first stage treats each row in $x(\cdot,\cdot\cdot)$ as a 1–D signal and compute its DTWS as described in

Section 2.5.2. The second stage then applies the DTWS to the resulting columns thereby computing the projections of $x(\cdot,\cdot)$ onto $\{V_{j-1}, W_{j-1}^{(1)}, W_{j-1}^{(2)}, W_{j-1}^{(3)}\}$.

2.5.4 Wavelet Packets

We now return to the treatment of 1–D functions and generalize the wavelet basis described in Section 2.5.2 to a family of orthonormal bases, called wavelet packet (WP) bases, that can be derived from a single orthonormal scaling function $\varphi(\cdot)$ and its associated wavelet $\psi(\cdot)$. The family of WPs $\{w(\cdot - k)\}_{n\in\mathbb{N},k\in\mathbb{Z}}$ is defined recursively by

$$w_0(\cdot) = \varphi(\cdot), \tag{2.45a}$$

$$w_1(\cdot) = \psi(\cdot), \tag{2.45b}$$

$$w_{2n}(t) = \sqrt{2}\sum_{k\in\mathbb{Z}} h^{(p)}[n]w_n(2t-k), \qquad \text{and} \tag{2.45c}$$

$$w_{2n+1}(t) = \sqrt{2}\sum_{k\in\mathbb{Z}} g^{(p)}[n]w_n(2t-k), \tag{2.45d}$$

where $2^p \le n \le 2^{p+1}$. We then have that for every partition $\mathcal{P}$ of $\mathbb{N}$ into sets of the form $I_{j,n} = \{n2^j, \ldots, (n+1)2^j - 1\}$ with $n, j \in \mathbb{N}$, the family of functions

$$\{w_n(2^j \cdot - k)\}_{k\in\mathbb{Z},I_{j,n}\in\mathcal{P}}$$

is an orthonormal basis for $\mathrm{L}^2(\mathbb{R})$. Each such basis can be associated with a tree–structured filter bank as the DTWS was associated with the octave band filter bank. As for the DTWS, the filter bank corresponding to a particular WP can be used to provide an efficient implementation of the decomposition and reconstruction of a sequence in the WP basis. Naturally, we can use these WPs for 2–D sequences using

the Kronecker product to construct the basis functions as in Section 2.5.3.

The rich family of WP bases afford us a multitude of signal representations in which to process a signal. We can thus choose to work within a *best basis* (BB) that is the optimal signal representation according to some criterion. For additive cost measures, there exist efficient tree pruning algorithms that find the BB for a particular signal [11]. Since entropy measures are additive (as defined in this context), this provides us with an efficient means of finding the lowest entropy signal representation of all wavelet packet representations. As discussed in Section 2.4, a low entropy representation is of crucial importance in data compression, since the entropy of a signal establishes a lower–bound on the degree to which it can, on the average, be losslessly compressed.

Chapter 3

Segmentation

Using the multiscale representation and scale AR models described in Chapter 2, we introduce in this chapter a technique for classifying pixels in SAR imagery via a statistical characterization of their evolution in scale. This technique has experimentally shown to be particularly well suited for classifying pixels according to the type of terrain surrounding them. It would thus be useful in applications where terrain information would be advantageous. One such instance is the use of segmentation in a preclassification phase of an ATR system that would automatically discard dense forest terrain based on the apriori knowledge that SAR is incapable of penetrating thick foliage and that a large target is unlikely to be located within a forest. An ATR system may further use the segmentation to focus attention near terrain boundaries where targets may attempt to conceal themselves. Compression, which is crucial to an operational SAR, can also use the segmentation to its advantage by exploiting specific statistical properties of various classes of terrain as discussed in Chapter 4.

The segmentation technique proposed here is inspired by that in [4] but in con-

trast, utilizes a model–based approach in place of an error residual examination. This modification reduces the size of the observation window required to obtain comparable statistical significance, and thereby increases the localization of the measurements. For each pixel at the finest resolution $\mathcal{I}_1[m,n]$, we generate an N-dimensional evolution vector $\mathbf{y}_{[m,n]}$ characterizing the local evolution in scale. A binary hypothesis test (BHT) is subsequently applied to $\mathbf{y}_{[m,n]}$ to classify $\mathcal{I}_1[m,n]$ as belonging to either a region of grass or of forest. Though this technique may clearly be generalized to include other classifications, we use grass and forest for their preponderance in a vast number of cases as the only two hypotheses for the sake of clarity.

In this chapter, we first describe how a measure of the evolution in scale of a pixel is obtained. We then statistically characterize this measure and use it in a binary hypothesis test to classify the pixel. Although this procedure can be followed for every pixel in the image to obtain the segmentation, we present an alternative approach that takes advantage of the strong correlation between the classification of neighboring pixels and as a result significantly reduces the computational costs. We then conclude this chapter by applying this technique to real SAR data and discussing the results.

3.1 Statistical Pixel Characterization

The procedure for classifying each pixel in the original SAR image is outlined in Figure 3.1. Upon generating an L-level multiscale quadtree representation for a SAR image, we proceed to obtain an N-dimensional evolution vector $\mathbf{y}_{[m,n]}$ to associate with each node in the finest scale image $\mathcal{I}_1[m,n]$. The evolution vector $\mathbf{y}_{[m,n]}$ consists of the scale AR modeling coefficients for a window of pixels centered around $\mathcal{I}_1[m,n]$

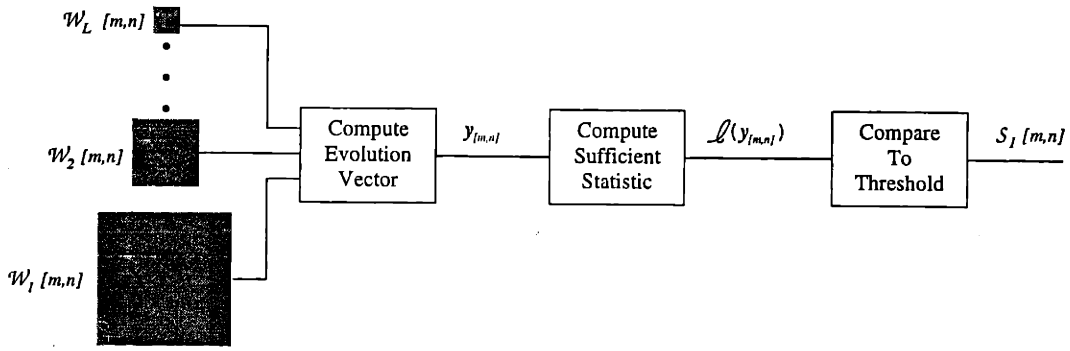


Figure 3.1: Diagram summarizing the steps involved in the segmentation technique.

and the corresponding ancestors on the quadtree, thus giving a measure of the *local* scale evolution behavior. We denote by $\mathcal{W}_1[m,n]$ the nodes in $\mathcal{I}_1$ contained within the $(2K+1)\times(2K+1)$ window centered at $\mathcal{I}_1[m,n]$ for some nonnegative integer K. The window of pixels in $\mathcal{I}_l$ corresponding to the ancestors of $\mathcal{W}_1[m,n]$ will similarly be denoted by $\mathcal{W}_l[m,n]$. The resulting AR coefficients modeling the evolution in scale of the set $\{\mathcal{W}_l[m,n]\}_{l=1}^{L}$ are the components constituting the vector $\mathbf{y}_{[m,n]}$ which we explicitly define next. The order of the regression, p_l, associated with modeling $\mathcal{W}_l[m,n]$ from its ancestors will vary with the level $l \in \{1,2,\ldots,L-1\}$ and is defined as

$$p_l = \begin{cases} L-l & \text{if } L-R \le l \le L-1, \\ R & \text{if } 1 \le l < L-R. \end{cases}$$

Such a definition ensures a maximal regression order at most equal to R and subject to the tree height. By applying Eq. (2.8) to $\mathcal{W}_l[m,n]$ for $l \in \{1,2,\ldots,L-1\}$, we

obtain the local scale AR model as

$$\hat{\mathbf{a}}_l[m,n] = \arg\left\{ \min_{\mathbf{a}_l} \left\{ \sum_{i=m-K}^{m+K} \sum_{j=n-K}^{n+K} \Big(\mathcal{I}_l[i,j] - a_{l,1}\mathcal{I}_{l+1}(\zeta_1[i,j]) - a_{l,2}\mathcal{I}_{l+2}(\zeta_2[i,j]) - \ldots - a_{l,p_l}\mathcal{I}_{l+p_l}(\zeta_{p_l}[i,j]) - \alpha_l \Big)^2 \right\} \right\}, \tag{3.1}$$

where we recall that $\mathbf{a}_l = [a_{l,1};\ a_{l,2};\ \ldots;\ a_{l,p_l};\ \alpha_l]$ and $\mathcal{I}_{l+i}(\zeta_i[i,j])$ represents the i^{th} ancestor of $\mathcal{I}_l[i,j]$. The regression vector $\hat{\mathbf{a}}_l[m,n]$ provides a statistically optimal description of the linear dependency of $\mathcal{W}_l[m,n]$ on $\{\mathcal{W}_k[m,n]\}_{k=l+1}^{l+p_l}$. The evolution vector $\mathbf{y}_{[m,n]}$ is subsequently obtained as

$$\mathbf{y}_{[m,n]} = \big[\mathbf{a}_1[m,n];\ \mathbf{a}_2[m,n];\ \ldots;\ \mathbf{a}_{L-1}[m,n]\big]. \tag{3.2}$$

Note that $\mathbf{y}_{[m,n]}$ is a vector reflecting the linear dependencies between the different resolutions in $\{\mathcal{W}_l[m,n]\}_{l=1}^{L}$, and from $\mathbf{y}_{[m,n]}$, we can generate the multiscale modeling parameters A and B in Eq. (2.4) for $\{\mathcal{W}_l[m,n]\}_{l=1}^{L}$. The evolution vector $\mathbf{y}_{[m,n]}$ thus provides a measure of the local evolution in scale.

When determining the size of the window used to compute the evolution vector, the tradeoff between *modeling consistency* and *local accuracy* must be taken into account. If too small a window is used, then the modeling may be too sensitive to variations due to the small number of nodes being utilized, particularly at the coarser resolutions. By increasing the window size, however, one increases the area contributing to the evolution vector and thus decreases the localization of the measurement. This effect would manifest itself near terrain boundaries where it may lead to misclassifications. Another drawback of larger window selection is the complication of mitigating edge effect problems due to insufficient data. One envisaged solution is to

use the largest possible window to compute the evolution vector or to use an extrapolation algorithm to extend the final segmentation map. This problem, however, is not a major concern since with reasonably sized windows, the edge regions constitute a negligible portion of SAR data.

3.2 Statistical Classification

A characterization of the evolution vector $\mathbf{y}_{[m,n]}$ is necessary to carry out a statistically meaningful classification of various types of terrain. Specifically, a BHT is applied to $\mathbf{y}_{[m,n]}$ to classify $\mathcal{I}_1[m,n]$ as a member of either a region of grass or of forest which are respectively designated as hypotheses H_g and H_f. In doing so, it is a simple matter to also classify $\mathcal{I}_l(\zeta_l[m,n])$ as a member of either region, thereby obtaining a multiresolution segmentation map. The procedure outlined here applies to each evolution vector $\mathbf{y}_{[m,n]}$. The classification of $\mathcal{I}_1[m,n]$ will depend only on $\mathbf{y}_{[m,n]}$ and the corresponding predetermined likelihoods $p\left(\mathbf{y}_{[m,n]}|H_g\right)$ and $p\left(\mathbf{y}_{[m,n]}|H_f\right)$. We thus omit the explicit spatial coordinates $[m,n]$ when clear from context.

To carry out a statistically significant hypothesis test, we need to specify the conditional probability density for $\mathbf{y}$ under each hypothesis. To do this, we extensively examined the distribution of evolution vectors obtained from a large homogeneous region of each of the terrains. For both grass and forest terrain, it turns out that each component in $\mathbf{y}$ is approximately Gaussian. We consequently make the approximation that $p(\mathbf{y}|H_g)$ and $p(\mathbf{y}|H_f)$ are N-variate Gaussian densities. They are then completely specified by their mean vectors $\mathbf{m}_g$ and $\mathbf{m}_f$ and their covariance matrices Λ_g and Λ_f all of which are calculated from the training data for each hypothesis. The special case of the ML detector for Gaussian likelihoods discussed in Section 2.3 of

Chapter 2 can then be used to classify each pixel resulting in the following decision rule

$$\ell(\mathbf{y}) = (\mathbf{y}-\mathbf{m}_F)^T\Lambda_F^{-1}(\mathbf{y}-\mathbf{m}_F) - (\mathbf{y}-\mathbf{m}_G)^T\Lambda_G^{-1}(\mathbf{y}-\mathbf{m}_G) \underset{\hat{H}=H_F}{\overset{\hat{H}=H_G}{\gtrless}} \log\frac{|\Lambda_G|}{|\Lambda_F|} = \lambda. \tag{3.3}$$

A preliminary segmentation map can then be formed by applying the above BHT to each evolution vector.

To address the issue of window size, we use a refinement procedure near terrain boundaries to enhance the segmentation as illustrated in Figure 3.2. The issue was that although larger windows produced accurate classifications within homogeneous regions, near boundaries, contaminated windows were apt to be misclassified due to their mixed statistical behavior. To address this problem, we first obtain a preliminary segmentation map $\mathbf{S}$ using a value of K sufficiently large to obtain a high level of statistical significance within homogeneous regions. In light of the high classification confidence level within these homogeneous regions, we proceed to refine the segmentation near boundary pixels by examining a telescopic sequence of windows centered at a common pixel. Since the window size is $(2K+1)\times(2K+1)$, any pixel whose window is not composed of homogeneous terrain will be within K pixels of the terrain boundary. Pixel $\mathcal{I}_1[m,n]$ thus only needs to be refined if the classification of each of the pixels in $\mathcal{W}_1[m,n]$ are not the same. To reduce the likelihood of window contamination for these pixels, we recompute their evolution vectors using a smaller value for K, e.g. $K/2$, and reapply the decision rule in Eq. (3.3). This procedure can then be iteratively applied until the desired degree of localization is achieved. In addition to reducing misclassifications due to window contamination, this procedure also increases the number of pixels in the interior of the image which may be pro-

cessed, i.e. those which are not too close to the edge of the image. It should be noted that although the mean and covariances for the grass and forest likelihoods may vary with K, their calculation imposes no additional computational costs on this technique since they can be determined beforehand.

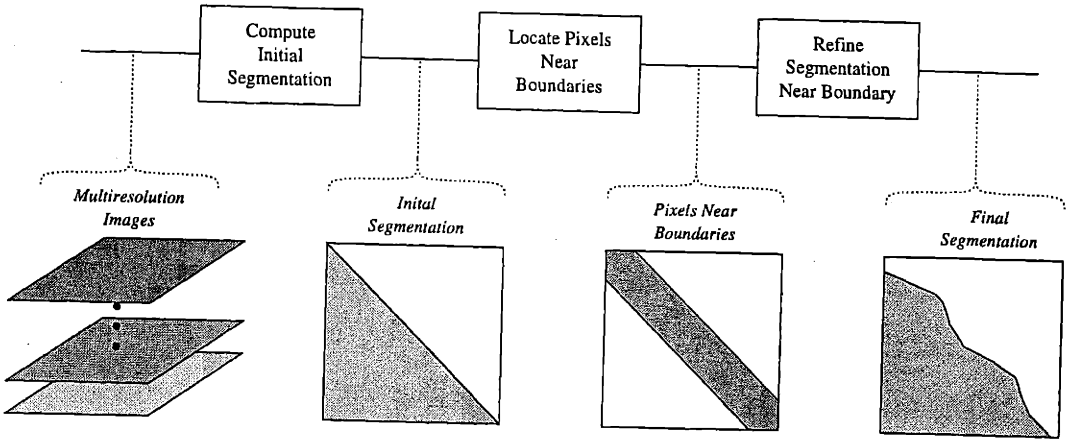


Figure 3.2: Illustration of boundary segmentation refinement process. An initial segmentation is computed to determine large homogeneous regions. The segmentation is then refined to localize the boundary location.

The procedure described above produces a segmentation of $\mathcal{I}_1$ which will be denoted by $\mathbf{S}_1[m,n]$. To determine the classification of $\mathcal{I}_l[m,n]$, denoted by $\mathbf{S}_l[m,n]$, the ML rule in Eq. (3.3) is summed over all nodes in $\mathcal{I}_1$ that have $\mathcal{I}_l[m,n]$ as an ancestor, i.e.

$$\sum_{[i,j]\left|\zeta_l[i,j]=[m,n]\right.} \ell\left(\mathbf{y}_{[i,j]}\right) \underset{\hat{H}=H_F}{\overset{\hat{H}=H_G}{\gtrless}} \sum_{[i,j]\left|\zeta_l[i,j]=[m,n]\right.} \lambda.$$

This can be interpreted as comparing the threshold λ to the average of sufficient

statistics and written as

$$\sum_{[i,j]\left|\zeta_l[i,j]=[m,n]\right.} \left(\lambda - \ell\left(\mathbf{y}_{[i,j]}\right)\right) \underset{\hat{H}=H_F}{\overset{\hat{H}=H_G}{\gtrless}} 0. \tag{3.4}$$

The decision rule is written as a comparison to zero because for a coarse resolution pixel $\mathcal{I}_l[m,n]$, the evolution vector of its descendants at the finest resolution may not have all been computed with the same window size. The value of λ thus also varies among those pixels since the covariances Λ_g and Λ_f vary with the window parameter K. Writing the decision rule as in Eq. (3.4), thus allows each test to be a comparison of the constant 0 to a quantity that can be computed from each evolution vector independently of its neighbors. This simplifies the comparison when classifying pixels whose descendants in $\mathcal{I}_1$ use different values of K in computing their evolution vectors.

Although averaging sufficient statistics does not yield a sufficient statistic for $\mathcal{I}_l[m,n]$, doing so is more advantageous than calculating a sufficient statistic as in Eq. (3.3) for each node at every level in the quadtree. One attribute of Eq. (3.4) is that the resulting segmentation map produced for $\mathcal{I}_l$ is consistent with the segmentation for $\mathcal{I}_1$ in the sense that if all the descendants of $\mathcal{I}_l[m,n]$ in $\mathcal{I}_1$ are classified as the same type of terrain, then $\mathcal{I}_l[m,n]$ will have that same classification. Averaging the finest resolution sufficient statistics also takes advantage of the scale–to–scale dependencies at the finer scales whereas computing a sufficient statistic from a truncated evolution vector ignores this information. Furthermore, truncation of the evolution vector would increase the sensitivity of the segmentation since the window used for $\mathcal{I}_l$ would contain $1/4^{l-1}$ as many pixels as the window for $\mathcal{I}_1$, thus lowering its statistical significance.

Note that the segmentation technique described here could easily be generalized to a larger number of terrain types. The choice to use grass and forest terrains as the only two hypotheses was motivated by the goal for a simple and clear illustration of our classification approach which can straightforwardly be extended to the M-ary hypothesis case.

3.3 Sparse Pixel Classification

Given the evolution vectors for two pixels that are close to each other, relative to K, their corresponding sufficient statistics should be nearly equal. We can use this fact to our advantage to improve the computational efficiency of our technique with negligible degradation in accuracy. Specifically, consider the sufficient statistics $\ell(y_{[m_0,n_0]})$ and $\ell(y_{[m_0+k_m,n_0+k_n]})$ generated for the pixels $\mathcal{I}_1[m_0,n_0]$ and $\mathcal{I}_1[m_0+k_m,n_0+k_n]$, respectively. Consider the case where both $|k_m|$ and $|k_n|$ are less than $2K+1$. The windows from which $y[m_0,n_0]$ and $y[m_0+k_m,n_0+k_n]$ are generated then overlap as illustrated in Figure 3.3. This overlapping of windows causes $\ell(y_{[m_0,n_0]})$ and $\ell(y_{[m_0+k_m,n_0+k_n]})$ to be correlated, particularly for smaller values of $\max\{k_m,k_n\}$. For instance, if the sufficient statistic is computed for every pixel, then for adjacent pixels, i.e. when $|k_m|+|k_n|=1$, we have that $\mathcal{W}_1[m_0,n_0]$ and $\mathcal{W}_1[m_0+k_m,n_0+k_n]$ have K^2-K out of a total of K^2 pixels in common resulting in their sufficient statistics being strongly correlated. One possible approach to reducing this redundancy, which we follow here, is to compute the evolution vector and its sufficient statistic for every κ pixels, i.e. for pixels $\mathcal{I}_1[\kappa m,\kappa n]$ as shown in Figure 3.4. To approximate the full image of sufficient statistics, simple bilinear interpolation can be applied to this downsampled image of sufficient statistics after which, thresholding can be applied to obtain the segmentation. Following such a procedure significantly reduces the number of evolution vectors

and sufficient statistics that need to be computed by a factor of κ^2, thus affording huge savings in computation.

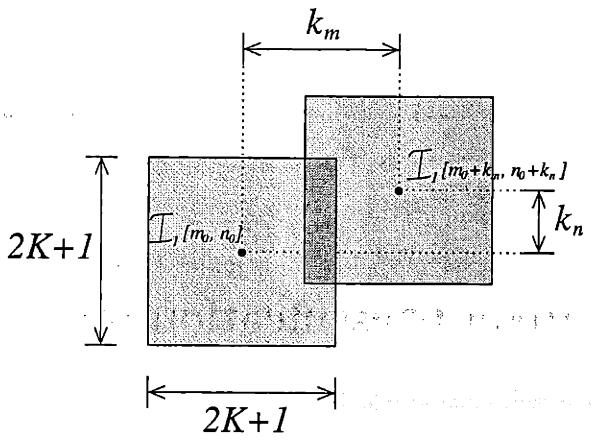


Figure 3.3: Illustration of overlapping windows for pixels near each other.

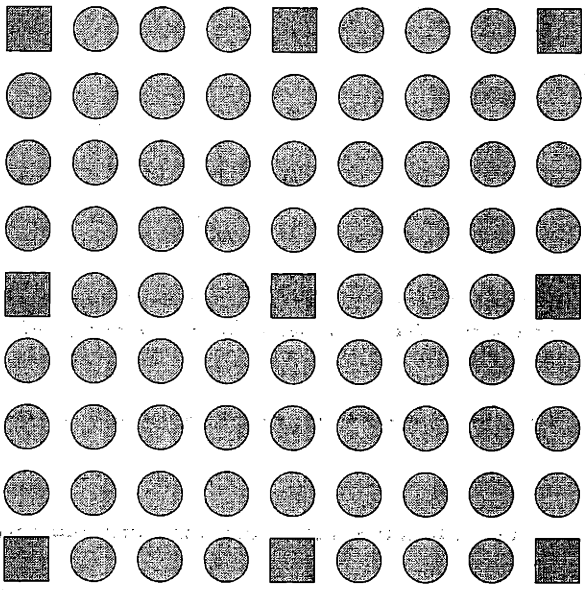


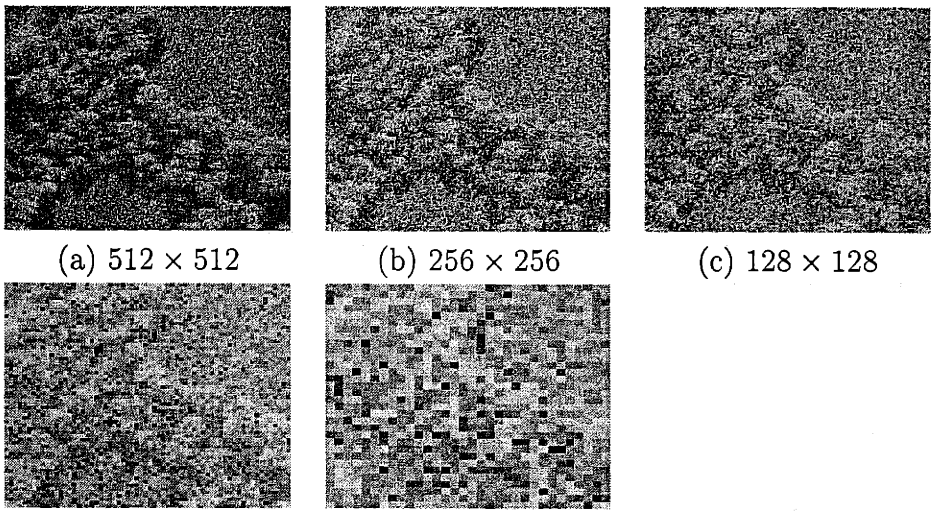
Figure 3.4: Illustration of sampling and interpolation grid with $\kappa = 4$. Square blocks represent the pixels for which sufficient statistics are computed. Circles represent pixels for which the sufficient statistics are approximated using bilinear interpolation.

3.4 Experimental Results

In this section, we apply our technique to data acquired with a 0.3 meter resolution HH polarization SAR using Lincoln–Laboratory's millimeter–wave SAR [12]. For each image, we generate a quadtree representation with $L = 5$ levels. A sequence of images $\{\mathcal{I}_l\}_{l=1}^{5}$ is shown in Figure 3.5. A maximum regression order of $R = 3$ was selected for both the grass and forest models. This was found to achieve a good tradeoff between modeling accuracy and computational efficiency. In fact, experimentation has shown that there is little gain in modeling accuracy for higher order models. Using these values for L and R results in an evolution vector $\mathbf{y}_{[m,n]}$ consisting of the coefficients

$$\mathbf{y}_{[m,n]} = [a_{1,1};\ a_{1,2};\ a_{1,3};\ \alpha_1;\ a_{2,1};\ a_{2,2};\ a_{2,3};\ \alpha_2;\ a_{3,1};\ a_{3,2};\ \alpha_3;\ a_{4,1};\ \alpha_4]\,,$$

parameterizing the evolution in scale of the windows $\{\mathcal{W}_l(\zeta_l[m,n])\}_{l=1}^{L}$.



(a) 512×512 (b) 256×256 (c) 128×128

(d) 64×64 (e) 32×32

Figure 3.5: A sequence of multiresolution images comprising the quadtree representation.

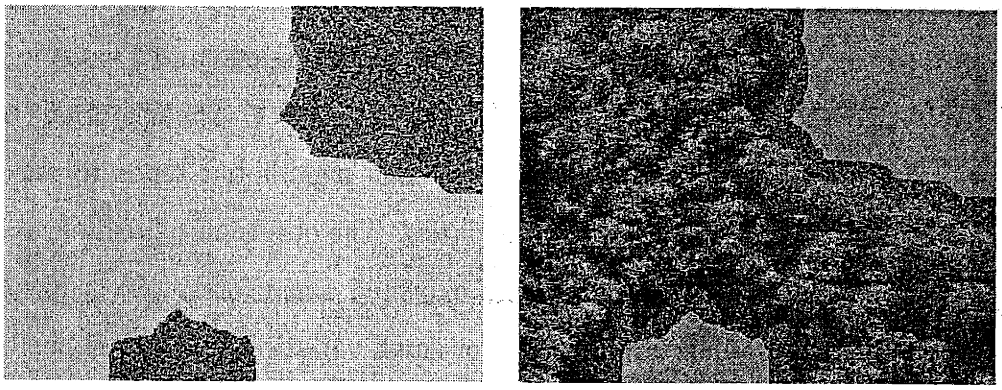
To specify the likelihoods $p(\mathbf{y}|H_g)$ and $p(\mathbf{y}|H_f)$, two 2048×2048 homogeneous images composed of grass and forest, respectively, were used as training data. The mean and covariance of the evolution vectors for the training data were then computed to determine the corresponding likelihoods. The mean vectors computed for $K = 16$ and $K = 32$, i.e. a 33×33 and a 65×65 window, are shown in Table 3.1. For each $a_{l,i}$, the coefficient corresponding to forest is larger than that to grass, thus indicating higher scale–to–scale correlation for forest terrain. This behavior is to be expected since, relative to forest terrain, grass terrain is composed of many small magnitude scatterers. As a result, a larger number of scatterers migrate into and out of a resolution cell as the scale is varied. One would thus expect grass terrain to exhibit less correlation in scale.

Table 3.1: Mean evolution vector for $K = 16$ and $K = 32$.

Terrain	Grass	Forest	Grass	Forest
K	16	16	32	32
$a_{1,1}$	0.3962	0.4539	0.3971	0.4695
$a_{1,2}$	0.0321	0.0701	0.0327	0.0770
$a_{1,3}$	0.0036	0.0425	0.0047	0.0483
α_1	-9.2794	-9.8511	-9.2764	-9.7952
$a_{2,1}$	0.2732	0.4042	0.2727	0.4350
$a_{2,2}$	0.0223	0.1029	0.0230	0.1177
$a_{2,3}$	0.0026	0.0633	0.0060	0.0658
α_2	-3.9988	-7.1523	-4.0314	-7.2930
$a_{3,1}$	0.2162	0.3862	0.2067	0.4207
$a_{3,2}$	0.0190	0.1097	0.0200	0.1224
α_3	1.9359	-2.3969	2.1035	-2.7548
$a_{4,1}$	0.2220	0.4103	0.1923	0.4227
α_4	7.0846	2.9838	7.8159	3.1190

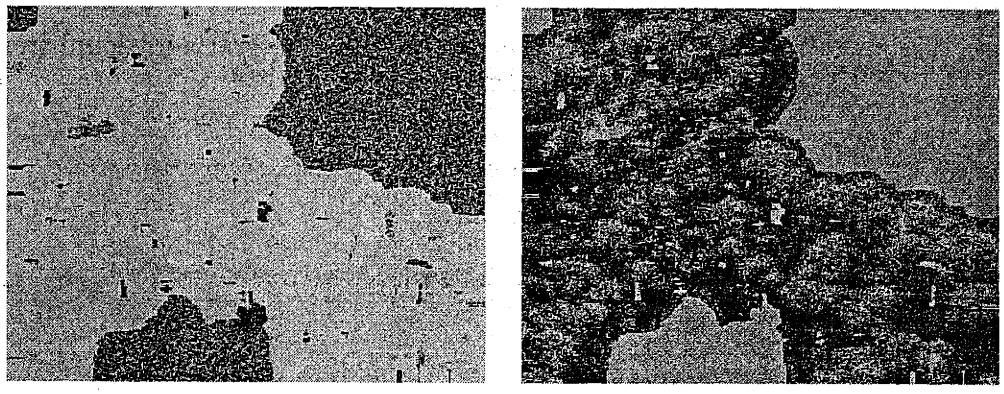
Figures 3.6 through 3.8 show the results of this segmentation technique on the image shown in Figure 3.5. To account for the pixels in $\mathcal{I}_1$ near the edges,

the segmentation map generated for the interior of the image is extended to the entire image by duplicating the edges of the segmentation map. Figures 3.6 and 3.7 show the results of applying the ML decision rule, without the boundary refinement procedure, to evolution vectors generated from 65×65 windows and 33×33 windows, respectively. The ML detector performs well for both cases. The use of the 65×65 window is extremely accurate within homogeneous regions but tends to misclassify grass pixels as forest near grass/forest boundaries. The bias to misclassify grass pixels near boundaries as forest terrain is to be expected since forest pixels have larger magnitude and thus produce larger errors for small variations in the AR coefficients. The coefficients generated by the least squares criterion are thus dominated by the few large magnitude forest pixels, as is the nature of least squares criteria. The use of the smaller 33×33 window reduces the number of errors associated with boundaries, and is however, slightly more sensitive to the variability within the homogeneous regions. For both window sizes, the segmentation accurately reflects the shape of the actual grass–forest boundary. Figure 3.8 displays the results of using one iteration of the refinement procedure to illustrate the added degree of localization which can be achieved without loss of consistency within homogeneous regions. Although only one iteration was performed, one could clearly perform several additional iterations to further improve localization.



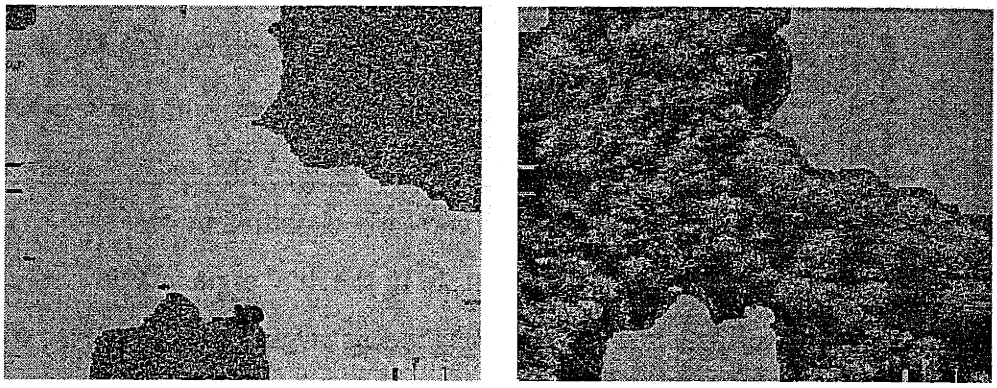
(a) Grass classifications. (b) Forest classifications.

Figure 3.6: Segmentation generated using a 65×65 window to generate the evolution vectors.



(a) Grass classifications. (b) Forest classifications.

Figure 3.7: Segmentation generated using a 33×33 window to generate the evolution vectors.



(a) Grass classifications. (b) Forest classifications.

Figure 3.8: Final segmentation utilizing both the 65×65 and 33×33 window segmentations.

The results of this segmentation technique were also applied to the intricate SAR image shown in Figure 3.9 containing two bright scatterers and small patches of grass surrounded by trees. A 65×65 window size was again used to generate an initial segmentation followed by boundary refinement with a 33×33 window size with results shown in Figures 3.10 through 3.12. The areas surrounding the two prominent scatterers are always classified as forest because they exhibit extremely strong correlation in scale. This strong correlation is to be expected, since each arises from a single scatterer whose magnitude is much larger than that of surrounding scatterers. As one varies the scale, the scatterers that migrate into and out of a given resolution cell thus have a negligible effect on its total magnitude which is dominated by the strong scatterer. Because forest regions demonstrate larger scale-to-scale correlation than grass regions, pixels whose window contains a strong scatterer are classified as forest. This then accounts for most of the $(2K+1) \times (2K+1)$ block around the scatterer as being classified as forest. Roads and large shaded areas also tend to be classified as forest as they possess a moderate scale–to–scale correlation which is more similar to forest than to grass.

The trade-off with the window size is particularly apparent in the lower–left region of the image where there is a small area of grass surrounded by trees. The 33×33 window performs significantly better in this region, and again displays its sensitivity in homogeneous regions. Even though much of the narrow path of grass on the middle–left portion of the image was correctly classified by the 33×33 window segmentation, it was designated as forest terrain in the final segmentation because the initial 65×65 window classified it as a homogeneous region of forest. This behavior is to be expected since the purpose of the window is to lessen the effects of highly localized statistical variations throughout the image. Due to the small size of the grassy clearing, shadows and trees contaminate the larger windows centered on those grass pixels, thus causing their misclassification.

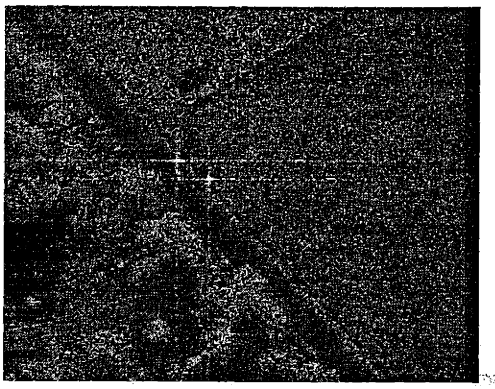
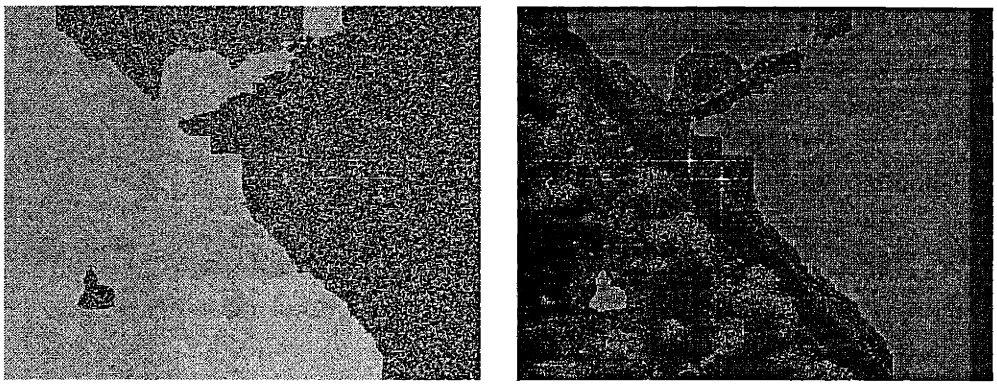
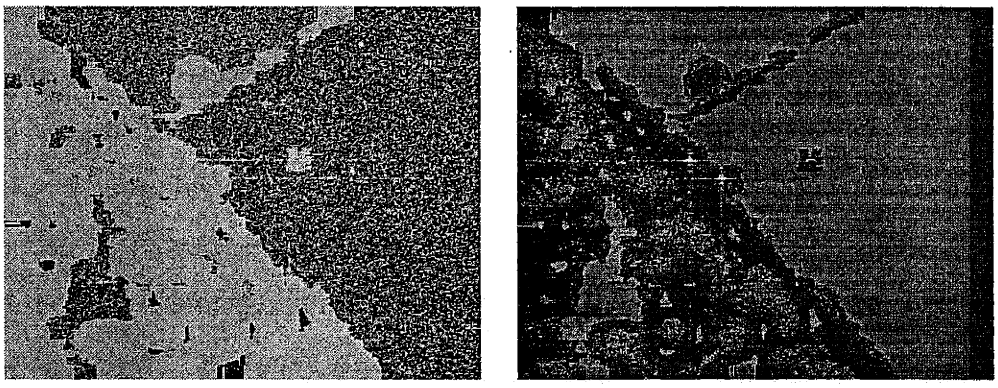


Figure 3.9: Finest scale image with two prominent scatterers.



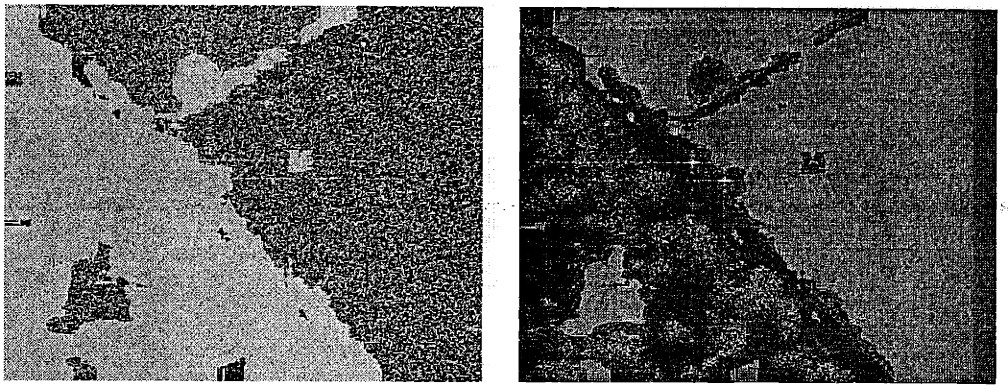
(a) Grass classifications. (b) Forest classifications.

Figure 3.10: Segmentation generated using a 65×65 window to generate the evolution vectors.



(a) Grass classifications. (b) Forest classifications.

Figure 3.11: Segmentation generated using a 33×33 window to generate the evolution vectors.

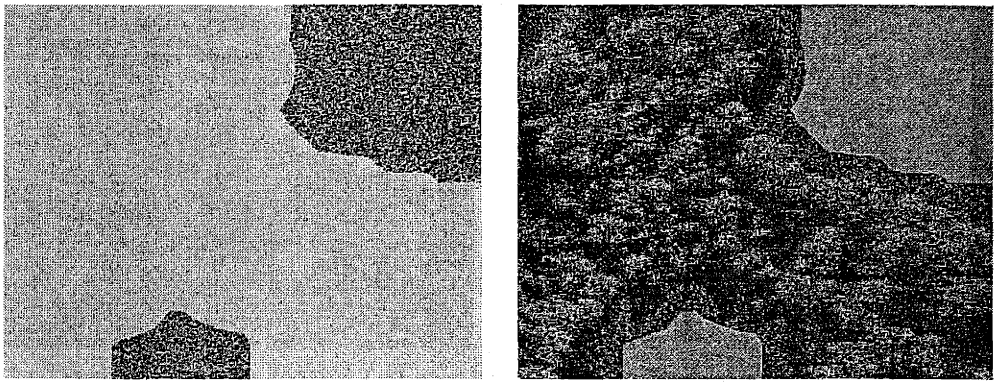


(a) Grass classifications. (b) Forest classifications.

Figure 3.12: Final segmentation utilizing both the 65 × 65 and 33 × 33 window segmentations.

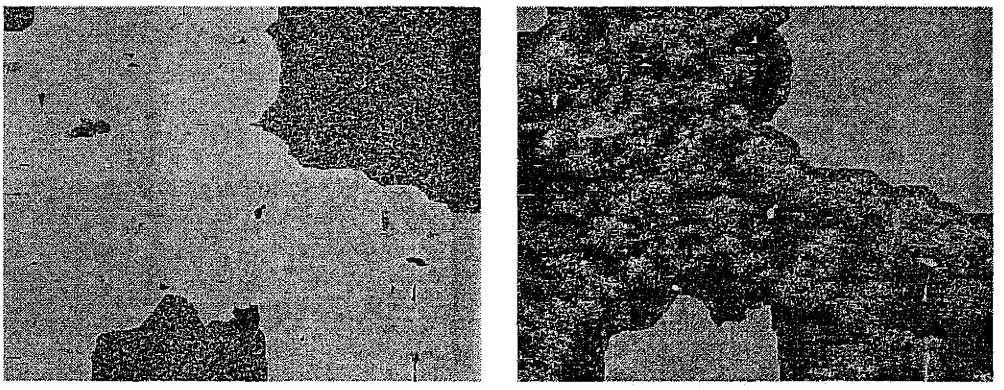
To demonstrate the efficacy of obtaining a segmentation from the interpolation of a sparse calculation of sufficient statistics, we apply this technique to the images shown in Figures 3.5a and 3.9. The initial segmentation uses a 65 × 65 window to calculate the sufficient statistic for the pixels $\{\mathcal{I}_1[\kappa m, \kappa n]\}_{m,n}$ with $\kappa = 16$. The remaining sufficient statistics are then approximated using bilinear interpolation and subsequently thresholded to generate the segmentation map. An identical procedure is followed to generate the sufficient statistics for pixels located in the vicinity of a boundary except that the sufficient statistics are then calculated using a 33 × 33 window for the pixels $\{\mathcal{I}_1[\kappa m, \kappa n]\}_{m,n}$ with $\kappa = 8$. For these values of κ=16 and 8, the number of evolution vectors and sufficient statistics to be computed decreases to 1/256 and 1/64 of that for the full sampled image, thus representing a tremendous savings in computation. The results for Figure 3.5a are shown in Figures 3.13 through 3.15; the results for Figure 3.9 are shown in Figures 3.16 through 3.18. In both cases, the interpolated sufficient statistics produced an almost identical segmentation as the fully sampled technique. The segmentation produced by sparse sampling and interpolation has more regular boundaries and more consistency within homogeneous regions. This is to be expected since subsampling and interpolating the image of

sufficient statistics is equivalent to lowpass filtering and thus suppresses some of the highly localized variations.



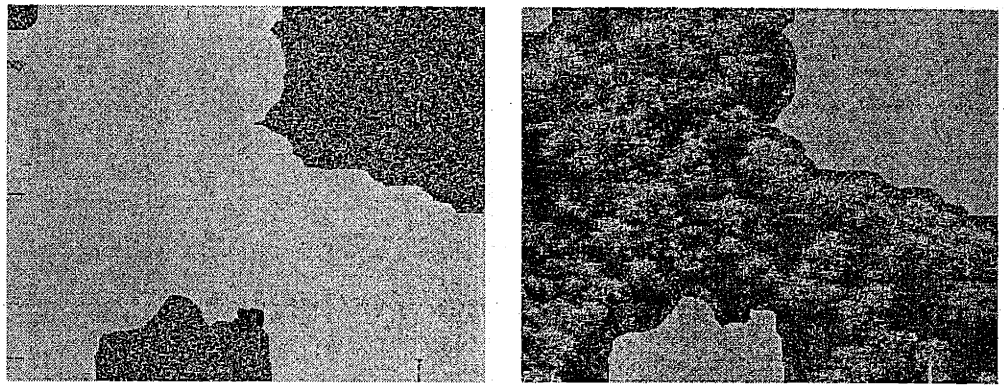
(a) Grass classifications. (b) Forest classifications.

Figure 3.13: Segmentation generated by interpolating the image of sufficient statistics calculated for $\mathcal{I}_1[16m, 16n]$ and using a 65×65 window.



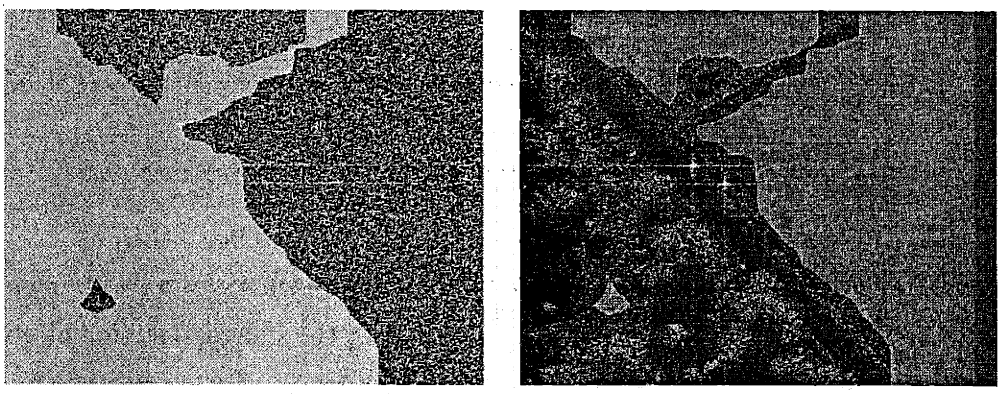
(a) Grass classifications. (b) Forest classifications.

Figure 3.14: Segmentation generated by interpolating the image of sufficient statistics calculated for $\mathcal{I}_1[8m, 8n]$ and using a 33×33 window.



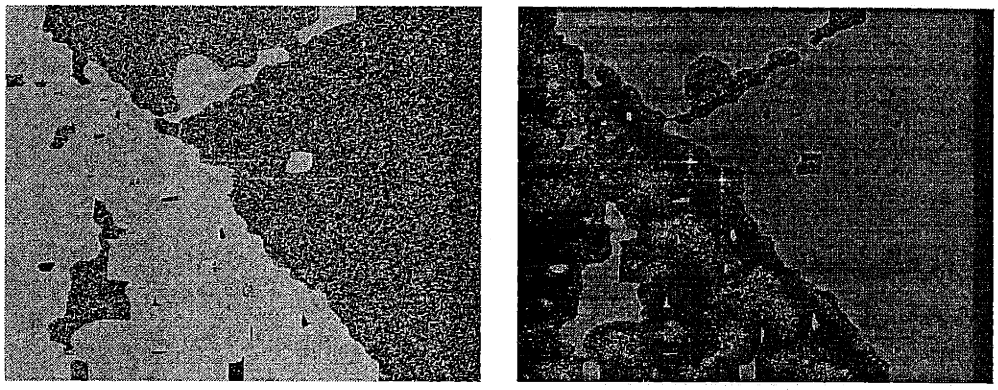
(a) Grass classifications. (b) Forest classifications.

Figure 3.15: Refined segmentation generated from the interpolated sufficient statistic images.



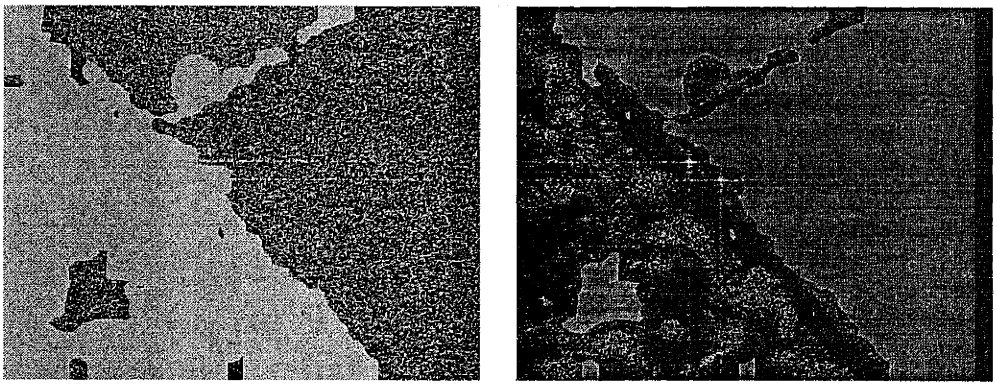
(a) Grass classifications. (b) Forest classifications.

Figure 3.16: Segmentation generated by interpolating the image of sufficient statistics calculated for $\mathcal{I}_1[16m, 16n]$ and using a 65×65 window.



(a) Grass classifications. (b) Forest classifications.

Figure 3.17: Segmentation generated by interpolating the image of sufficient statistics calculated for $\mathcal{I}_1[8m, 8n]$ and using a 33×33 window.



(a) Grass classifications. (b) Forest classifications.

Figure 3.18: Refined segmentation generated from the interpolated sufficient statistic images.

Chapter 4

Multiresolution Image Compression

The amount of data acquired by a SAR system can be staggering. As mentioned in Chapter 3, segmentation can be used to judiciously discard nonrelevant imagery based on apriori knowledge of the situation at hand. Although this may reduce much of the data that needs to be intensively examined, there will most likely remain a significant amount of data that requires further investigation. To abate the costs associated with storing/transmitting this information, we present in this chapter a multiresolution segmentation–directed compression (SDC) algorithm as a natural extension of the segmentation in Chapter 3 and of the multiscale framework.

The cornerstone of the SDC is the prediction of fine resolution images from coarser ones using scale AR models. As described in Chapter 3, these models are used to generate a set of multiresolution segmentation maps that classify pixels according to terrain by measuring the evolution in scale on a quadtree. By utilizing these seg-

mentation maps in tandem with the models used to generate them, we remove much of the locally redundant information between different resolutions. This prediction is used within a pyramid encoder which initially codes the segmentation and the coarse decimated image $\mathcal{I}_L$ and then predicts the finer resolutions. In order to compensate for prediction errors, it also codes the prediction error residuals for each resolution prior to predicting any finer resolutions. To further improve the coding efficiency, the error residuals are transformed using optimized wavelet packet bases and soft–thresholded prior to quantization. These quantized coefficients are then losslessly compressed using arithmetic coding. The use of this hierarchical coding procedure allows for multiresolution decoding at the receiver.

In this chapter, we first describe the procedure used to encode the multiresolution segmentation maps. We then present a pyramid coder that takes advantage of the segmentation and terrain models to minimize the redundancies in the representation. In the following two subsections, we specifically focus on the prediction of finer resolutions and on the coding of the error residuals. We finally conclude with the presentation of experimental results demonstrating the effectiveness of this technique.

4.1 Encoding the Segmentation Map

To take full advantage of local redundancies in the imagery, both the encoder and decoder make use of the the multiresolution segmentation maps, thus necessitating their coding before the multiresolution images $\{\mathcal{I}_l\}_{l=1}^{L}$. The finest resolution segmentation map $\mathbf{S}_1$ is coded first by processing the pixel classifications in a raster scan format with an arithmetic coder similar to that described in Section 2.4 of Chapter 2. This arithmetic coder is adaptive to the conditional probability den-

sity $p(\mathbf{S}_1[m,n]|\mathbf{S}_1[m-1,n],\mathbf{S}_1[m,n-1])$, thus taking advantage of the strong dependence of each pixel's classification on that of its neighbors. This strong dependence is a consequence of a segmentation map most likely consisting of large regions of constant value that reflect the homogeneous regions of terrain comprising a SAR image. It is thus very unlikely that $\mathbf{S}_1[m,n]$ will be different from both $\mathbf{S}_1[m-1,n]$ and $\mathbf{S}_1[m,n-1]$, thereby yielding near zero probabilities for all classifications different from $\mathbf{S}_1[m-1,n]$ and $\mathbf{S}_1[m,n-1]$. Defining states by $(\mathbf{S}_1[m-1,n],\mathbf{S}_1[m,n-1])$ thus allows the arithmetic coder to take advantage of knowledge of the classification of surrounding pixels. To further reduce redundancies, the estimated conditional probability distribution for each state adapts to the history of outcomes that occur when in that state. The conditional distribution for each state should thus converge to the true underlying conditional distribution if it is stationary. Using an arithmetic coder in this respect has experimentally shown to provide significantly better compression than both run–length coding and the JBIG algorithm.

The nodes in the intermediate resolution segmentation map $\mathbf{S}_l$ can usually be determined from $\mathbf{S}_1$ as a result of the construction of the coarser resolution maps. If all the descendants of $\mathbf{S}_l[m,n]$ in $\mathbf{S}_1$ share the same classification, the averaging of sufficient statistics in $\mathbf{S}_1$ guarantees a similar classification for $\mathbf{S}_l[m,n]$. Because SAR images are almost always composed of large homogeneous regions of terrain, most of the pixels in $\mathcal{I}_l$ will have descendants with a common classification that can be determined from $\mathbf{S}_1$. The decoder determines the classifications in $\mathbf{S}_l$ that can be ascertained from $\mathbf{S}_1$. In order to obtain an exact reconstruction of $\mathbf{S}_l$, the encoder needs only to code those classifications in $\mathbf{S}_l$ that cannot be determined from $\mathbf{S}_1$. It should be noted that when coding the nondeterministic classifications, neither the level l nor the spatial coordinates $[m,n]$ need to be specified. Such overhead is unnecessary since the encoder can anticipate the inability of the decoder to determine the classification of a node if they process the nodes in the same spatial order. We

specifically use a lexicographical ordering where the nodes at each level are processed in a raster scan format.

4.2 Pyramid Encoding

Following the encoding of the segmentation maps, we proceed to code the set of multiresolution images $\{\mathcal{I}_l\}_{l=1}^{L}$ using a pyramid–based coding algorithm. The intrinsic structure of our quadtree analysis is naturally adapted to pyramid encoding and affords through the image representation, a simple and efficient coding technique. The segmentation–directed stochastic modeling of the varying image resolutions in our technique is one of the key properties that differentiates it from classical pyramid coding. The segmentation partitions the quadtree into regions characterized by a particular evolution in scale, thus allowing the stochastic properties of each class of terrain to be more fully exploited in the compression.

Given the pyramid of multiresolution images, and assuming that the coarser resolutions $\mathcal{I}_{l+1}, \mathcal{I}_{l+2}, \ldots, \mathcal{I}_L$ have already been coded, we illustrate in Figure 4.1 the principles involved in coding the next resolution $\mathcal{I}_l$. The predicted image, $\widehat{\mathcal{I}}_l$, for resolution l is computed using the segmentation map $\mathbf{S}_l$ and the previously reconstructed coarser resolution images $\{\widetilde{\mathcal{I}}_k\}_{k>l}$. The error image $\mathbf{E}_l$ is then computed as the difference between the original image $\mathcal{I}_l$ and the predicted image $\widehat{\mathcal{I}}_l$, i.e. $\mathbf{E}_l = \mathcal{I}_l - \widehat{\mathcal{I}}_l$. At the decoder, knowledge of the error residual $\mathbf{E}_l$ is sufficient to reconstruct $\mathcal{I}_l$ as it can just as well compute the predicted image $\widehat{\mathcal{I}}_l$. By coding $\mathbf{E}_l$ in place of $\mathcal{I}_l$, the compression efficiency is increased since the dynamic range of $\mathbf{E}_l$ is reduced from that of $\mathcal{I}_l$, hence lowering the redundancy within homogeneous regions of terrain. For a further gain in compression efficiency, $\mathbf{E}_l$ is transformed into a domain with a rapidly

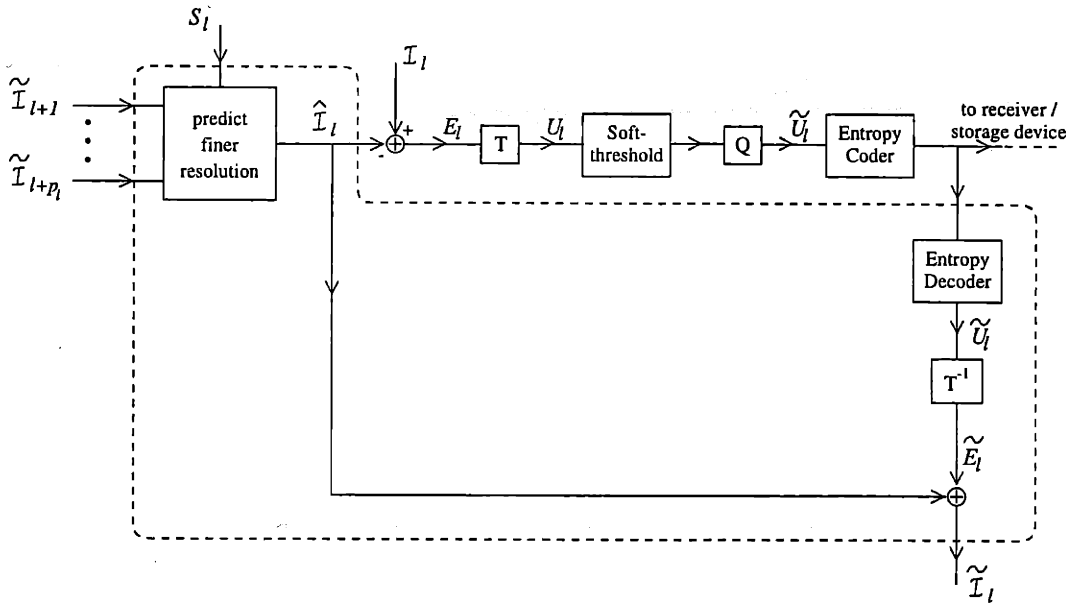


Figure 4.1: Illustration of the iterative procedure used to encode finer resolutions. The dashed region represents operations performed to decode the image.

decaying coefficient distribution. The transformed image $\mathbf{U}_l$ is then soft–thresholded to remove the speckle noise in the image in addition to now providing an even more parsimonious representation. The image is then quantized to produce $\widetilde{\mathbf{U}}_l$ which is entropy coded prior to being transmitted to the decoder.

The structure within the dashed boundary in Figure 4.1 reconstructs the image $\mathcal{I}_l$ by using the corresponding segmentation map, the previously reconstructed coarser resolution images, and the residual data sent to the decoder. The quantized error data $\widetilde{\mathbf{U}}_l$ is inverse transformed to generate the reconstructed error image $\widetilde{\mathbf{E}}_l$. The reconstructed image, $\widetilde{\mathcal{I}}_l$, at resolution l is then obtained by adding the predicted image $\widehat{\mathcal{I}}_l$ to the reconstructed error residual $\widetilde{\mathbf{E}}_l$, i.e. $\widetilde{\mathcal{I}}_l = \widetilde{\mathbf{E}}_l + \widehat{\mathcal{I}}_l$. This reconstruction is performed at both the encoder and decoder to preserve consistency in the two predicted images for all future iterations. As a result, the error residual remains exactly the difference between the actual image and the predicted image at the decoder. Reconstructing the images at both ends thus prevents a possible compounding of coding

errors from different resolutions.

Another attribute of this coding technique is that the totality of the coding error in the reconstruction of the current resolution l is limited to that arising from the soft-thresholding and quantization at that resolution, i.e. errors in the reconstruction of coarser resolutions cannot limit the reconstruction quality of the current resolution. To explain the origin of this robust behavior, we note that any errors in the reconstruction of coarser resolutions will be reflected as errors in the predicted image. Since the error residual corrects all prediction errors, the error in the reconstructed image $\widetilde{\mathcal{I}}_l$ is then equal to the error incurred by coding the error residual $\mathbf{E}_l$, all of which occurs in the soft–thresholding and quantization steps. This then provides a technique which is robust to unexpected evolution in scale due to any source such as objects that conform to neither the grass nor forest model.

The pyramid structure additionally affords a hierarchical data coding capability, where the decoder can use an initial portion of the transmitted data to reconstruct the small coarse image at resolution L. After reconstructing the image at any particular resolution $l \in \{2, \ldots, L\}$, the decoder predicts a finer resolution $l-1$. It then uses the next portion of the data stream, i.e. the coded error residual, to compensate for prediction errors and refine the detail in resolution $l-1$. While this procedure is repeated until the image has been reconstructed at its finest resolution, it is not necessary to wait for all of the data to be transmitted before reconstructing the images. Each resolution is available immediately after the corresponding error residual data has been received. The transmission and subsequent reconstruction of a resolution can thus be potentially utilized in a feedback loop where a "request to stop transmission" of the current image may be issued by the decoder any time it deems advantageous.

4.2.1 Prediction of Finer Resolutions

The pyramid coder predicts $\mathcal{I}_l$ from the segmentation map $\mathbf{S}_l$ and the previously reconstructed coarser resolution images $\widetilde{\mathcal{I}}_{l+1}, \widetilde{\mathcal{I}}_{l+2}, \ldots, \widetilde{\mathcal{I}}_{p_l}$. The equation for predicting $\mathcal{I}_l$ is obtained by rewriting Eq. (2.7) and is given by

$$\widehat{\mathcal{I}}'_l[m,n] = b_{l,\mathbf{S}_l[m,n]} + \sum_{i=1}^{p_l} a_{l,i,\mathbf{S}_l[m,n]} \widetilde{\mathcal{I}}_{l+i}(\zeta_i[m,n]), \tag{4.1}$$

where $b_{l,\mathbf{S}_l[m,n]}$ and the $a_{l,i,\mathbf{S}_l[m,n]}$ are the parameters used to predict $\mathcal{I}_l[m,n]$ from the reconstructions $\widetilde{\mathcal{I}}_{l+i}(\zeta_i[m,n])$ under the AR model indicated by $\mathbf{S}_l[m,n]$. The use of segmentation–directed modeling allows the minimization of statistical redundancies between scales, specifically, those associated with terrain classes. We are thus able to obtain a more accurate modeling of the evolution in scale for the entire image. This results in a better prediction which in turn, decreases the variance of the error residuals thus yielding higher coding efficiency.

To compensate for the blocking artifacts associated with the prediction in the quadtree structure, $\widehat{\mathcal{I}}'_l$ is filtered to produce $\widehat{\mathcal{I}}_l$, the predicted image for resolution l. The filter used here is the Kronecker product of 1–D filters whose coefficients are obtained from the values of a Gaussian probability density. We define $x = \sqrt{\frac{10}{\mathcal{L}_{l-1}}}$ where $\mathcal{L}_{l-1}$ is the number of quantization levels used to code the previous resolution's error residual which will be defined in Section 4.2.2. The coefficients of the 1-D filter can then be written as the values of a normalized Gaussian probability density sampled at $[-3x, -2x, -x, 0, x, 2x, 3x]$. Using this dynamic filter adjusts the degree of smoothing according to the coarseness of the quantization. For large $\mathcal{L}_{l-1}$, x is small and the filter does very little smoothing. As the number of quantization levels decreases, the lowpass behavior of the filter increases. Using this filter has experimentally shown that the quasi–elimination of the blocking artifacts is achieved

without sacrificing the details in the images.

4.2.2 Error Residual Coding

Most of the resources in the encoded data are taken up by the details in the prediction error residual for each of the resolutions. Efficient compression of these residuals is thus essential to the efficiency of the algorithm as a whole. In order to further improve compression performance, each error residual is transformed using a globally optimized wavelet packet basis unique to its resolution. Because the wavelet packet basis for each resolution is apriorily known, computational and transmission/storage resources are conserved in that the BB search and coding are obviated. These resolution–specific optimized bases are determined in advance by computing the best–basis (BB) representation for a large set of error residual training data. Each wavelet packet is optimized in the sense that it produces the minimal Shannon–Weaver entropy[1] of the resulting representation for the training data as described in [11]. We thereby consider the entropy of a generic residual image minimized. These bases will henceforth be referred to as characteristic–basis (CB) wavelet packets to differentiate them from the actual BB wavelet packets for arbitrary error residuals. The BB representation of a signal possesses a rapidly decaying exponential distribution with a maximal number of zero coefficients [13]. Because the CB is a close approximation of the BB for a given resolution, the CB similarly produces a rapidly decaying distribution of coefficients and nearly a maximal number of zero coefficients. In fact, of all wavelet packet bases, the CB will on the average produce the most zero coefficients in the transformed error residuals since the statistics of the training set are representative of those of the residuals for the corresponding resolution.

[1] the definition of the Shannon–Weaver entropy was given in Eq. (2.15)

Upon taking the CB transform of an error residual, a soft–threshold is applied to all the transformed coefficients. Doing so skews the distribution of coefficients toward zero and decreases its entropy while preserving much of the detail, thereby improving compression performance. Soft–thresholding CB coefficients also has a denoising effect akin to that of soft–thresholding small BB coefficients. This allows us to remove much of the speckle inherent in SAR imagery. The value of the threshold t_l for resolution l is given by

$$t_l = \sigma_l \sqrt{2 \log (n_l)}, \tag{4.2}$$

where n_l is the number of pixels in $\mathbf{E}_l$ and σ_l is the standard deviation of the speckle noise in the original image $\mathcal{I}_l$. Because the speckle arises out of the construction of the multiresolution images, it is *independent* of the coding process. Assuming that the speckle noise is white and that $\mathcal{I}_l$ has negligible energy at higher frequencies, we proceed to estimate σ_l as the sample standard deviation of the coefficients in the high–high frequency subband of the standard wavelet decomposition of $\mathcal{I}_l$ as proposed in [14].

Following the soft–thresholding procedure, the coefficients are uniformly quantized. The number of quantization levels used for each resolution is dynamically determined. It is a function of the compression parameter, "q", which is specified at run–time and which thereby controls the extent of the compression for the image. The number of quantization levels, $\mathcal{L}_l$, used for resolution l is computed as,

$$\mathcal{L}_l = 1 + 2 * rnd \left(\frac{q}{10} * \frac{\sigma_{\mathbf{E}_l}}{\sigma_{\mathcal{I}_l}} \right),$$

where $\sigma_{\mathbf{E}_l}$ is the standard deviation of the error residual, $\sigma_{\mathcal{I}_l}$ is the standard deviation of the original image at resolution l, and $rnd(\cdot)$ rounds a number to the nearest integer. It is clear that $\mathcal{L}_l$ has been constrained to be an odd number. This allows

the quantizer to have symmetrically arranged quantization levels including one at zero. The argument of the rounding operation is proportional to the quality factor "q" and inversely proportional to the signal–to–noise ratio (SNR), with the noise taken as the prediction error rather than the speckle noise in $\mathcal{I}_l$ that was discussed earlier. For a fixed value of "q", a particular image quality is thus maintained at each resolution as measured in terms of SNR. Although this allocation among the resolutions has worked well in experimentation, the optimality of quantization level allocation is a topic of further investigation.

Following quantization, the coefficients are arithmetically coded to take full advantage of their low entropy. The arithmetic coder is similar to the one used for coding the segmentation map $\mathbf{S}_1$ in that it uses probabilities conditioned on adjacent pixels to exploit the correlation within each subband. Given that the coefficients corresponding to higher frequencies are usually much smaller than those in the lowest subband, we separately code the lowest frequency subband and its complement. This results in a more efficient coding of the higher frequency subbands due to their higher likelihood of zero coefficients in comparison to the lowest frequency subband. This allows, in turn, the arithmetic coder to converge to their underlying distribution more quickly.

4.3 Experimental Results

As for the results presented for the segmentation technique, to demonstrate the SDC, we apply it to data gathered with a 0.3 meter resolution HH polarization SAR using Lincoln–Laboratory's millimeter–wave SAR. For each 512×512 image, a quadtree with $L = 5$ levels is generated and a maximal regression order of $R = 3$ is used to

model the evolution in scale. For all the results in this section, the segmentation described in Chapter 3 is used. The initial segmentation is constructed using a 65×65 window by computing the sufficient statistic for every $\kappa = 16$ pixels in the horizontal and vertical direction and interpolating the results. The segmentation was then refined near boundaries using a 33×33 window to compute the sufficient statistic for every $\kappa = 8$ pixels in the horizontal and vertical directions and interpolating the results. The model parameters used for the prediction in Eq. (4.1) are taken as the parameter means given in Table 3.1 when using a 33×33 window.

We first demonstrate the claim of near optimality regarding the entropy in the CB representation. Figure 4.2 shows fifteen 512×512 SAR images for which prediction error residuals are generated under perfect quantization. Each of the error residuals are transformed with the BB, CB, and standard wavelet transform (WT). For each of these transforms, we use a maximal wavelet packet tree depth of four levels. For each of the images, Table 4.1 displays the percent deviation between the Shannon–Weaver entropy of the CB representation and that of the BB representation. Table 4.2 similarly gives the percent deviation between the WT representation entropy and that of the BB representation. In all cases, the entropy of the CB transformed residual is nearly optimal, particularly at the finer resolutions. The only instances in which the wavelet transform yielded a lower entropy are for error residuals at the fourth level of the quadtree where the high sensitivity of the BB to the small number of nodes is apparent. As expected, the entropy of the CB approaches that of the BB and outperforms the wavelet transform as one proceeds to finer resolutions, which is of more significance due to the increased number of pixels.

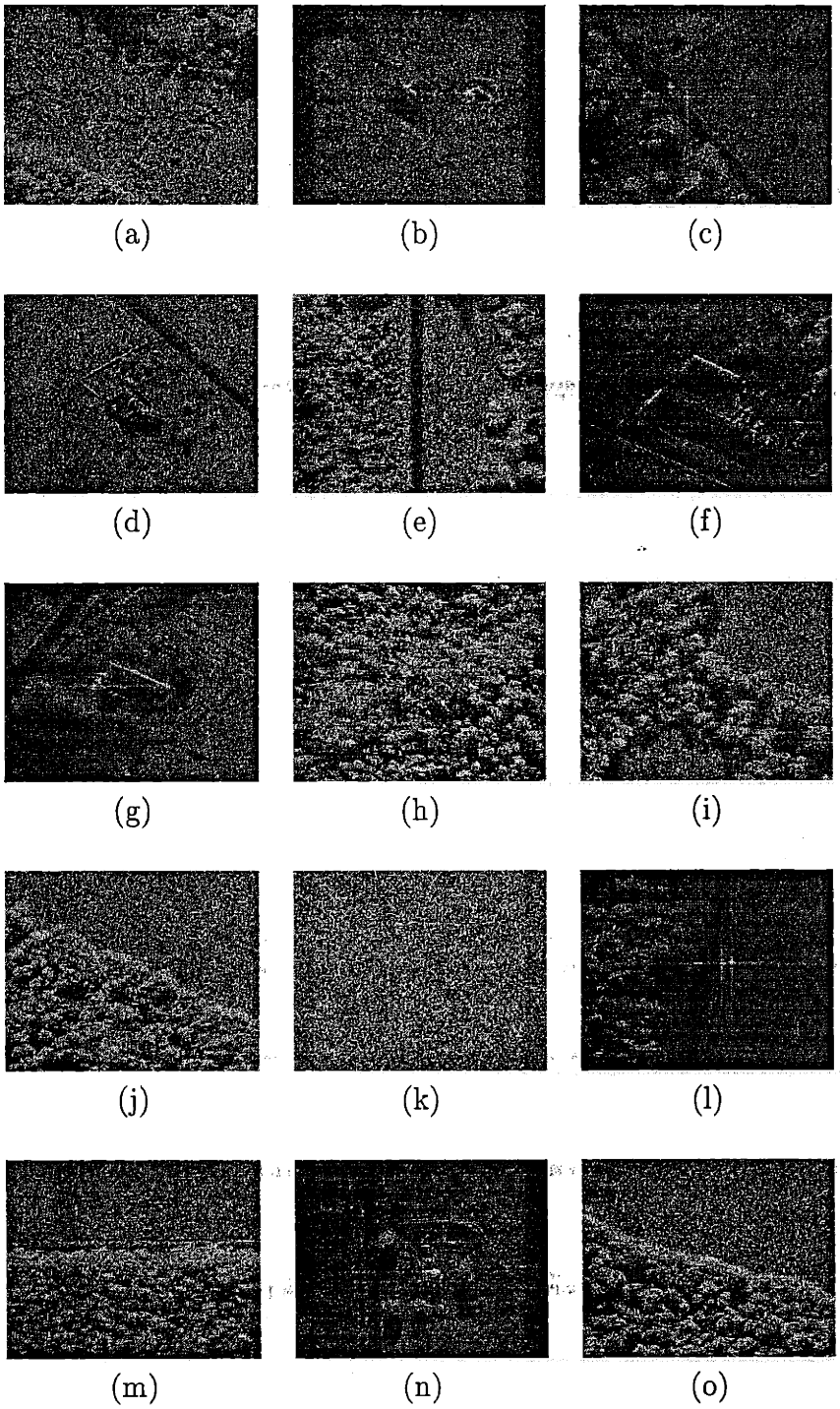


Figure 4.2: SAR images used to generate error residuals to which the entropies in the BB, CB, and wavelet transform representation are compared.

Table 4.1: Percent deviation of CB transformed entropy from BB transformed entropy for test images.

Figure 4.2	E_1	E_2	E_3	E_4
(a)	0.10 %	0.05 %	0.39 %	1.12 %
(b)	0.11 %	0.13 %	0.50 %	1.41 %
(c)	0.14 %	0.04 %	0.26 %	1.58 %
(d)	0.07 %	0.18 %	0.28 %	1.25 %
(e)	0.13 %	0.09 %	0.36 %	0.85 %
(f)	0.12 %	0.12 %	0.40 %	1.09 %
(g)	0.11 %	0.17 %	0.38 %	0.83 %
(h)	0.18 %	0.11 %	0.27 %	0.88 %
(i)	0.12 %	0.07 %	0.39 %	1.10 %
(j)	0.12 %	0.14 %	0.25 %	0.85 %
(k)	0.11 %	0.18 %	0.78 %	1.68 %
(l)	0.10 %	0.22 %	0.29 %	0.92 %
(m)	0.14 %	0.10 %	0.54 %	0.66 %
(n)	0.08 %	0.16 %	0.33 %	0.94 %
(o)	0.12 %	0.07 %	0.11 %	1.09 %
average deviation	0.12 %	0.12 %	0.46 %	1.30 %
maximum deviation	0.18 %	0.22 %	0.78 %	1.68 %

Table 4.2: Percent deviation of wavelet transformed entropy from BB transformed entropy for test images.

Figure 4.2	E_1	E_2	E_3	E_4
(a)	1.01 %	1.08 %	3.33 %	3.52 %
(b)	0.30 %	0.78 %	1.51 %	0.61 %
(c)	0.90 %	1.13 %	1.64 %	3.14 %
(d)	0.30 %	0.87 %	1.75 %	2.58 %
(e)	0.98 %	0.90 %	2.06 %	0.90 %
(f)	0.70 %	1.11 %	2.37 %	2.08 %
(g)	0.59 %	1.55 %	2.86 %	2.67 %
(h)	1.16 %	0.47 %	0.79 %	1.56 %
(i)	1.37 %	1.13 %	1.98 %	3.49 %
(j)	0.94 %	0.82 %	1.78 %	2.15 %
(k)	0.18 %	0.40 %	0.88 %	1.22 %
(l)	0.63 %	0.69 %	1.87 %	3.86 %
(m)	1.34 %	0.72 %	2.04 %	2.84 %
(n)	0.57 %	0.43 %	1.09 %	0.76 %
(o)	1.34 %	0.57 %	1.32 %	2.82 %
average deviation for WT	0.77 %	0.71 %	1.63 %	2.15 %
maximum deviation	1.37 %	1.55 %	3.33 %	3.86 %

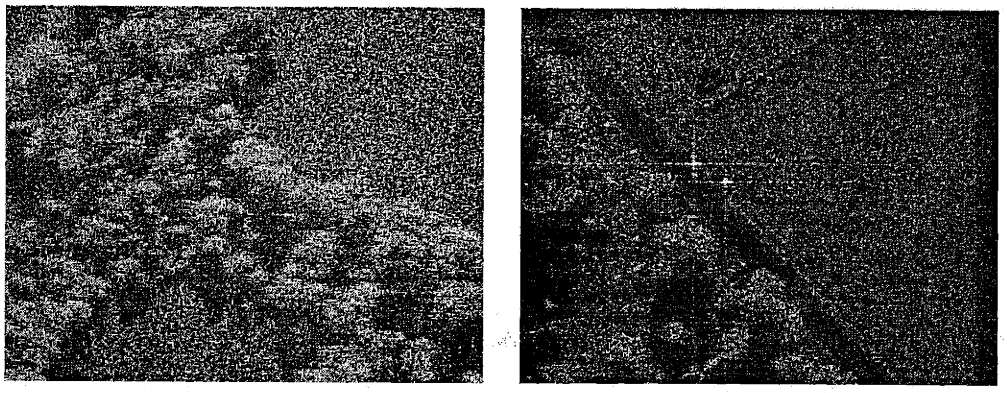
The SDC is applied to the SAR images that are used to demonstrate the segmentation in Chapter 3. We display the log–magnitude of these two images again in Figure 4.3. Both of the original complex images have a precision of 24 bits per pixel (bpp); 12 bpp for each of the real and of the imaginary components. Figures 4.4 and 4.5 show how the SDC compares to the Embedded Zerotree Wavelet (EZW)[15] and JPEG[16] algorithms at compressing the images in Figure 4.3. Table 4.3 displays the peak signal–to–noise ratios (PSNR) of the reconstructed images, which is defined as

$$\text{PSNR} = 10\log_{10}\left(MN\frac{\left(\max_{[m,n]}\{\mathcal{I}_1[m,n]\} - \min_{[m,n]}\{\mathcal{I}_1[m,n]\}\right)^2}{\sum_{[m,n]}\mathbf{E}_1^2[m,n]}\right),$$

where M and N are the dimensions of the finest resolution image and $\mathbf{E}_1$ is the error in the reconstructed image at the finest resolution. To obtain a fair comparison of all the techniques, we subtract the number of bytes used to code the segmentation from our compressed file size. We do this because JPEG and EZW have no associated segmentation, which, as previously noted, is necessary in many applications. For such applications, it would thus be necessary to either independently code the segmentation or to store/transmit the entire complex imagery which will later be segmented. The extent to which JPEG can compress the images is limited as reflected by the lower compression performance compared to SDC and EZW. In spite of its lower compression ratio, JPEG still performed considerably worse than the other two. The SDC and EZW perform fairly well at this extremely high compression ratio. EZW tends to produce more "mosquito" artifacts than the SDC. On the other hand, the SDC tends to smooth the image more than EZW, particularly when the soft–thresholding is disabled. The better performance with the soft–thresholding comes as a result of the more rapidly decaying distribution it produces, which yields higher compression in the entropy coding stage. The two anomalies in Figure 4.5 are preserved quite well by both the SDC and EZW. Note that this is absolutely essential for many applica-

tions such as ATR. The robustness of the SDC is demonstrated by its effectiveness in handling these two bright scatterers which conform to neither the grass nor the tree models in its evolution in scale.

The quality of the reconstructed images produced by the SDC could be increased by the addition of model hypotheses, particularly an "anomaly" model that corresponds to the high correlation in scale produced by strong scatterers. The existence of such a model would allow the segmentation to detect these anomalous regions and the compression to more accurately predict their evolution in scale. With this particular classification, the segmentation offers the additional use as a preliminary anomaly detector at no extra cost.

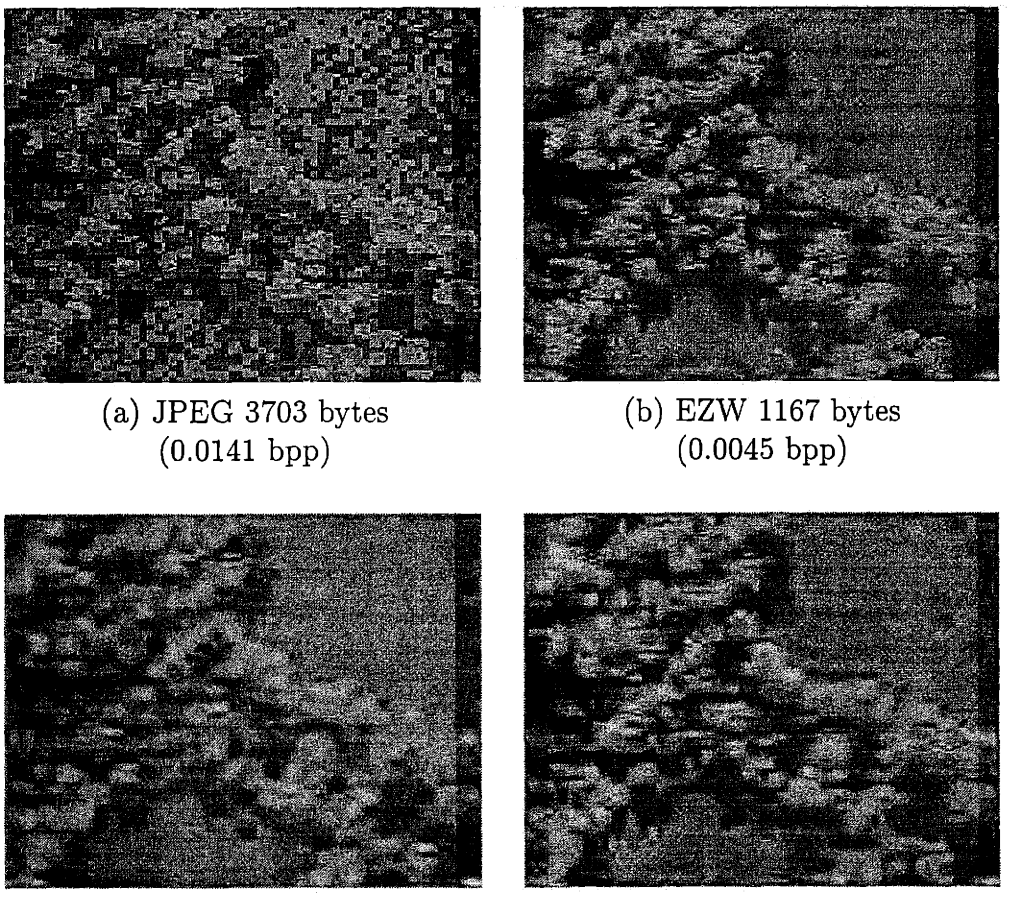


(a) (b)

Figure 4.3: Log–magnitude of finest resolution images to which the compression technique is applied. (a) "Typical" image composed of forest and grass. (b) Image containing two bright anomalies.

Table 4.3: PSNR comparing DMM, EZW, and JPEG compression.

Original Image	JPEG	EZW	SDC (no soft–thresh.)	SDC (soft–thresh.)
Figure 4.3(a)	14.46 dB	20.65 dB	18.40 dB	18.79 dB
Figure 4.3(b)	22.47 dB	24.16 dB	23.67 dB	23.87 dB



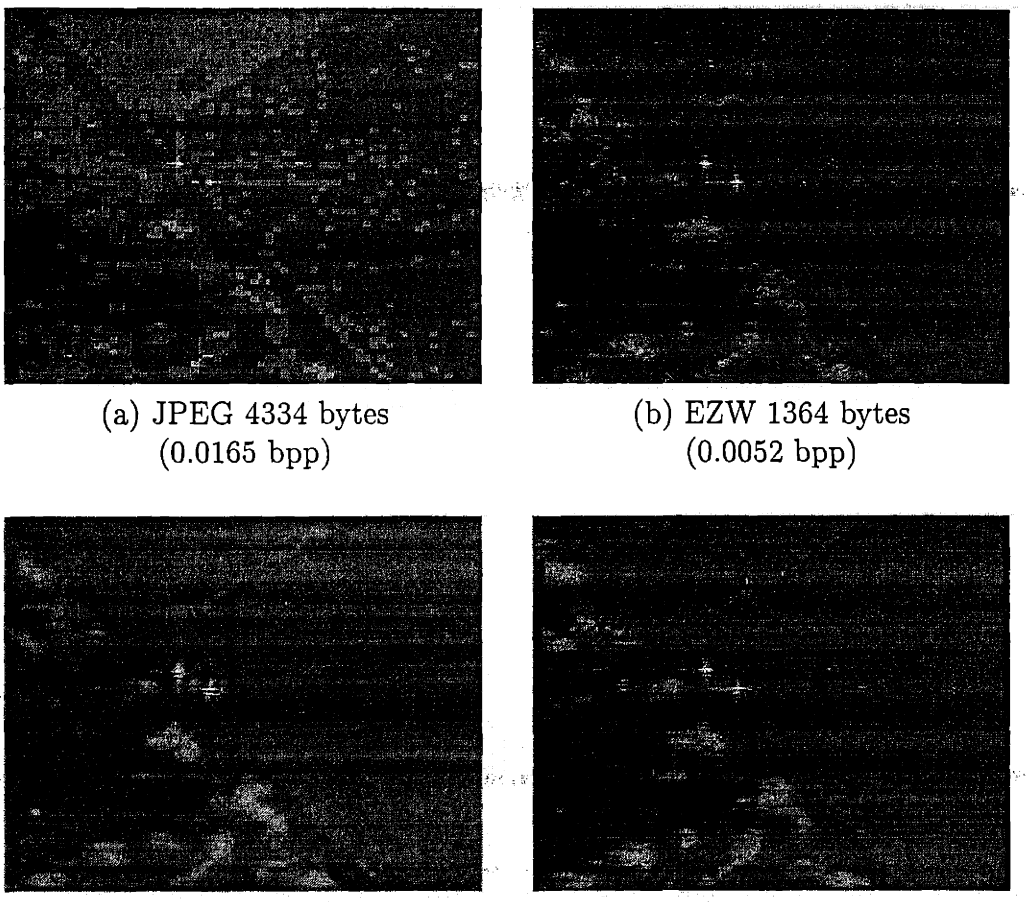
(a) JPEG 3703 bytes (0.0141 bpp)

(b) EZW 1167 bytes (0.0045 bpp)

(c) SDC (no soft–thresh.) 988 bytes (0.0038 bpp)

(d) SDC (soft–thresh.) 1092 bytes (0.0042 bpp)

Figure 4.4: Comparison of JPEG, EZW, and SDC compression techniques applied to Figure 4.3a.



(a) JPEG 4334 bytes (0.0165 bpp)

(b) EZW 1364 bytes (0.0052 bpp)

(c) SDC (no soft–thresh.) 1051 bytes (0.0040 bpp)

(d) SDC (soft–thresh.) 1217 bytes (0.0046 bpp)

Figure 4.5: Comparison of JPEG, EZW, and compression techniques applied to Figure 4.3b.

Chapter 5

Conclusions and Extensions

In this chapter, we will briefly summarize and provide concluding remarks for the content of Chapters 3 and 4. This will then be followed by possible extensions of the work presented here.

5.1 Conclusions

In this thesis, we have utilized hierarchical modeling within a multiscale representation of SAR imagery for segmentation and subsequent compression of the log–magnitude image. The efficacy of this technique arises from the accurate modeling of terrain classes as scale AR stochastic processes. We first summarize here the contributions presented and then provide concluding remarks.

5.1.1 Terrain Segmentation

We obtain the terrain classification of a pixel according to the evolution in scale of a window surrounding that pixel. This approach has shown to be quite accurate when used to distinguish between grass and forest terrains in SAR imagery. Terrain segmentation offers utility in many applications in remote sensing by highlighting areas of interest warranting further analysis. We used the segmentation for the image compression technique described in Chapter 4.

To perform the segmentation, we use a multiscale quadtree representation of the image. The Markov property of multiscale representations affords highly parallelizable algorithms offering fast analysis of the imagery. In addition to affording fast algorithms, the multiscale quadtree offers a framework in which the evolution in scale of terrain is well captured by scale AR models. With each terrain class, we identify a particular scale AR model. To measure the evolution in scale around a pixel, we compute an evolution vector as the best scale AR model describing the evolution in scale of a window. These evolution vectors are viewed as random variables and modeled as N–variate Gaussians with a particular mean and covariance both of which are conditioned on the terrain of the corresponding region. With this statistical characterization, we apply a ML decision rule to classify the center pixel. In performing the ML decision rule, we generate a sufficient statistic from the evolution vector and compare it to a threshold. Judicious utilization of these sufficient statistics permits significant reductions in computation. The sufficient statistics of pixels near each other are strongly correlated because of the overlapping windows from which their evolution vectors are computed. To take advantage of this correlation, we compute the evolution vectors and corresponding sufficient statistics for a subsampled set of points. We then interpolate these sufficient statistics to approximate those for the remaining points. Following such a procedure affords reductions in computation by

two orders of magnitude while creating no noticeable artifacts in the accuracy of the segmentation.

We compute the evolution vector from a sequence of multiresolution windows sufficiently large to achieve statistical significance. If too small of a window is used, then the likelihood densities of the evolution vectors conditioned on each hypothesis, overlap due to their high variance. On the other hand, using too large of a window reduces the locality of the measurement and leads to misclassifications near boundaries. To achieve both the high statistical significance associated with larger windows and the locality associated with smaller windows, we use a refinement procedure. The segmentation is first performed using a window sufficiently large to obtain classifications with high statistical significance in homogeneous regions of terrain. For this segmentation the actual terrain boundary will be within the window size of the computed boundary. We can thus repeat the segmentation process with a smaller window over a small group of pixels that are near the previously computed boundary to refine its true location. This refinement may be further iterated with smaller windows until the desired degree of boundary localization is achieved. Because the refinement involves only those few pixels near boundaries, this procedure imposes little additional computational costs.

5.1.2 Compression

SAR image compression addresses the problem of huge costs incurred by storing or transmitting the vast amount of data gathered by SAR systems. Compression increases the efficiency in data acquisition in a single SAR mission by increasing storage capacity or by allowing immediate transmission to a ground station.

The compression technique described in Chapter 4 takes advantage of the segmentation and hierarchical models by using them in tandem to remove redundancies inherent to pixels of the same terrain class. To do this, it first codes the coarsest resolution image and segmentation map. It then follows an iterative procedure to code finer resolutions. This procedure progressively supplies to the decoder the fine details that arise by proceeding to a finer resolution which the multiscale modeling cannot predict. In order to efficiently code these details, we apply a CB wavelet packet transform that is optimized to efficiently represent the characteristics of the error residual for each particular resolution. These coefficients are then soft–thresholded to remove the speckle in the imagery and produce a more rapidly decaying coefficient distribution. This is followed by quantization and entropy coding.

The performance of this technique is shown to remarkably outperform JPEG and to be equivalent to that of EZW. Unlike our technique, neither JPEG nor EZW provide a segmentation of the imagery that could be utilized in subsequent analysis. We also demonstrated the ability of our technique to preserve anomalous structures at extremely high compression ratios which is of paramount importance in applications such as ATR.

5.2 Extensions

Motivated by the results presented here, we describe other possible avenues of future research utilizing the multiscale framework for SAR image compression.

5.2.1 Compressing the phase information

Although the log–magnitude SAR image has shown to be reliable for target detection, there are still many algorithms that make use of both the magnitude and the phase information. The primary obstacle to compactly coding the phase information has been the difficulty in modeling its erratic behavior. One would expect there to be little correlation among the phase of a pixel at varying resolutions for forest terrain (and even less for grass terrain) due to the coherent averaging of a large number of radar returns. We would thus not expect to be able to highly compress the phase without considerable loss of quality. We would expect bright scatterers, however, to display some correlation in their phase among scales due to the presence of the small number of very large magnitude reflections. In applications where these bright scatterers are the only object of interest, such as ATR, it may then be sufficient to compress the phase to a moderate degree, yielding high quality reconstructions for strong scatterers and fair reconstructions for natural terrain.

5.2.2 Segmentation Directed Bit Allocation

If the reconstruction quality of some terrain classifications are more important than other classifications, then the segmentation map may be used to adjust the bit allocation accordingly. For instance, if we use three classifications (grass, forest, and bright scatterer) for the purpose of ATR, then we may choose to assign significantly more bits to those pixels classified as bright scatterers which may possibly contain targets.

Bibliography

[1] U. Benz, K. Strodl, and A. Moreira, "A Comparison of Several Algorithms for SAR Raw Data Compression," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 33, pp. 1266–1276, Sep. 1995.

[2] R. Baxter and M. Seibert, "SAR Image Compression with the Gabor Transform: A Comparison of Different Quantizers and Bit Allocation Methods," in *Proc. of the SPIE, Algorithms for SAR Imagery III*, Apr. 1996.

[3] W. W. Irving, "Multiresolution Approach to Discriminating Targets From Clutter SAR data acquiredin SAR Imagery," in *SPIE Symposium*, Apr. 1995.

[4] C. Fosgate, H. Krim, W. Irving, W. Karl, and A. Willsky, "Multiscale Segmentation and Anomaly Enhancement of SAR Imagery," *IEEE Trans. on Image Proc.*, vol. 6, pp. 7–20, Jan. 1997.

[5] D. Munson and R. Visentin, "A Signal Processing View of Strip-Mapping Synthetic Aperture Radar," *IEEE Trans. on Acoustics, Speech, and Sig. Proc.*, vol. 37, pp. 2131–2146, Dec. 1989.

[6] M. Basseville, A. Benveniste, K. Chou, S. Golden, R. Nikoukhah, and A. Willsky, "Modeling and Estimation of Multiresolution Stochastic Processes," *IEEE Trans. on Info. Theory*, vol. 38, pp. 766–784, Mar. 1992.

[7] T. Cover and J. Thomas, *Elements of Information Theory.* New York: John Wiley and Sons, Inc., 1991.

[8] I. Witten, R. Neal, and J. Cleary, "Arithmetic Coding for Data Compression," *Comm. of the ACM*, vol. 30, pp. 520–540, Jun. 1987.

[9] G. Langdon and J. Rissanen, "Compression of Black–White Images with Arithmetic Coding," *IEEE Trans. on Comm.*, vol. 29, pp. 858–867, Jun. 1981.

[10] A. Cohen and J. Kovacevic, "Wavelets: The Mathematical Background," *Proceedings of the IEEE*, vol. 84, pp. 514–522, Apr. 1996.

[11] M. Wickerhauser, "Lectures on Wavelet Packet Algorithms," Nov. 1991.

[12] J. C. Henry, "The lincoln laboratory 35ghz airbourne polarimetric sar imaging system," in *IEEE National Telesystems Conference*, p. 353, Mar. 1991.

[13] H. Krim, "On the Distributions of Optimized Multiscale Representations," in *ICASSP '97*, (Munich, Germany), IEEE, Apr. 1997.

[14] D. Wei, H. Guo, J. E. Odegard, M. Lang, and C. S. Burrus, "Simultaneous speckle reduction and data compression using best wavelet packet bases with application to SAR based ATD/R," in *Proc. of the SPIE, Mathematical Imaging: Wavelet Applications for Dual Use*, Apr. 1995.

[15] J. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Trans. on Sig. Proc.*, vol. 41, pp. 3445–3462, Dec. 1993.

[16] G. K. Wallace, "The JPEG Still Picture Compression Standard," *Comm. of the ACM*, vol. 34, pp. 30–44, Apr. 1991.

[17] P. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. on Comm.*, vol. COM-31, pp. 532–540, Apr. 1983.

[18] A. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[19] J. Lim, *Two-Dimensional Signal and Image Processing.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[20] N. Hess-Nielsen and M. Wickerhauser, "Wavelets and Time–Frequency Analysis," *Proceedings of the IEEE*, vol. 84, pp. 523–540, Apr. 1996.

[21] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding.* Englewood Cliffs, NJ: Prentice Hall, 1995.

[22] ISO-IEC/JTC1/SC2/WG9, *Progressive Bi-level Image Compression*, Apr. 1992.