1995

Multiresolution Techniques in Image Processing.

Ramana L. Rao
Louisiana State University and Agricultural & Mechanical College

MULTIRESOLUTION TECHNIQUES IN IMAGE PROCESSING

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Computer Science

by
Ramana L. Rao
B.Tech., Indian Institute of Technology, Madras, 1988
M.S., Louisiana State University, Baton Rouge, Louisiana, 1994
May 1995

Acknowledgements

I thank Prof. S.S. Iyengar for chairing my doctoral committee and providing uncommon support during my tenure at the Robotics Research Laboratory. I also thank Dr. Don Kraft, Dr. Si-Qing Zheng, Dr. Jianhua Chen, Dr. Frank Neubrander and Dr. J. Ramanujam for serving on my doctoral committee.

It is a pleasure to acknowledge Dr. George Zweig in the Theoretical Division of the Los Alamos National Laboratory. Without his indulgence, it would have been extremely difficult to complete this dissertation. A special word of thanks is also due to the Center for Nonlinear Studies for supporting part of this research.

I would like to thank Jim Bollman and Rob Buckley of Xerox Corporation for supporting me on research internships at the Joseph C. Wilson Research Center in Webster, New York, during the summers of '93 and '94.

Thanks are also due to Ron Holyer and Matthew Lybanon of the Naval Research Laboratory, John C. Stennis Space Center, for providing satellite images for testing some of the techniques described in this dissertation.

I would like to thank Lakshman Prasad, Ajit Pendse, Amit Nanavati and Dr. Daryl Thomas for many enlightening discussions. Dr. Vibhas Aravamuthan has been a constant source of technical advice, encouragement and words of wisdom throughout my graduate life. He has always been extremely generous with his time and effort and I am truly fortunate to have him as a friend.

Subbiah Rajanarayanan, Vinayak Hegde, Sankar Krishnamurthy, Raghu Yedatore, Sundar Vedantham and Harish Manohara, among others, have been a frequent source of moral support.

Contents

List of Tables

List of Figures

Abstract

Multiresolution, an effective paradigm for signal processing, offers a natural, hierarchical view of information. As a computational tool, multiresolution can be applied to a variety of problems in signal and image processing. For instance, feature detection and extraction can be performed quickly and efficiently using a multiresolutional method to analyze images.

This dissertation addresses the problems of segmentation and edge detection in images with particular emphasis on satellite images which display features of a fine nature, such as the Atlantic Gulf Stream. The dissertation also describes a portable toolkit, for conventional and multiresolutional image processing, that was developed to test the various algorithms described in this research. The dissertation is motivated by the importance of the problems of segmentation and edge detection in the area of image processing, and the ever-present need for effective, efficient algorithms.

Chapter 1

Introduction

In recent years, multiresolution techniques are increasingly being applied to image processing problems. Global features are quickly and efficiently extracted from images through these techniques. However, the idea of analyzing images at different *scales* of *resolution* is not new (see [61]). Research in several fields such as mathematics, physics, signal processing, and seismic data analysis has a common foundation in wavelets and multiresolution analysis. The most interesting characteristic of wavelets is their simplicity. A single function is translated and scaled to generate a basis for analyzing all functions of a certain class.

Multiresolution offers an efficient framework for extracting information from images at various levels of resolution [61, 43]. Several applications in diverse fields use this idea. Pyramid algorithms (e.g. see [5]) use a multiresolution architecture to reduce the computational complexity of image processing operations. Gauch and Pizer [20] describe a multiresolution approach to image feature analysis. Liu and Yang [40] use multiresolution for segmenting color images. Lifshitz and Pizer [38] propose a hierarchical segmentation technique using intensity extrema in images. We have used the idea of multiresolution decomposition to efficiently integrate information in distributed sensor networks [51, 53].

The idea of multiresolution is not necessarily wavelet-related; the so-called *pyramid algorithms* [3, 6, 69, 61] incorporate a multiresolution strategy that is not wavelet-based.

1.1 Organization

This dissertation is organized as follows. Chapter 2 introduces the notation and mathematical preliminaries necessary for the remainder of the dissertation. Chapter 3 describes the discrete wavelet transform for one and two dimensional signals. This transform is fundamental to some of the techniques developed in this dissertation.

Chapter 4 describes the segmentation algorithm based on multiresolution histogram decomposition. The segmentation scheme analyzes the histogram of a gray scale image using the idea of multiresolution. This approach is different from most conventional multiresolution techniques in image processing in that, the technique is applied not to the image data but to the image histogram. The overall algorithm is linear in the size of the image. The algorithm is flexible and does not make any assumptions regarding the modality of the histogram of an image.

Chapter 5 addresses the problem of edge detection — the conceptual dual of segmentation. Two algorithms are presented and their performance on images with weak edges is compared. Both these algorithms rely on the segmentation algorithms developed in Chapter 4. To place the results obtained using these algorithms in perspective, the performance of conventional edge detectors such as the derivative operator and the morphological gradient operator is also shown.

Chapter 6 describes *Spectrum* — a portable, extensible toolkit for conventional and multiresolution image processing developed as a testbed for the research described in this dissertation. Chapter 7 summarizes the dissertation and indicates possible research directions for the future.

The bibliography at the end of the dissertation is comprehensive and is intended to be a good starting point for any literature search in the areas of wavelets,

multiresolution and image processing. Although it is comprehensive at the time of this writing, it will soon be dated considering the fact that wavelets and multiresolution is a research area that is undergoing rapid changes. More current on-line sources such as *CARL* and the World-Wide Web page *www.math.scarolina.edu* are recommended for up-to-date information about this area.

In the remainder of this dissertation, the words 'signal' and 'image' will be used interchangeably. Thus, a signal may be either a one-dimensional entity or one that is two-dimensional. Where we restrict our attention to the one-dimensional case, we will clearly qualify it.

1.2 Image Processing

An *image* is a two-dimensional array of numbers. These numbers may be integers or real numbers. Usually, these numbers are *quantized* to a finite number of values. A *pixel* is the smallest element in an image. It represents an atom in the image and serves as a place marker for the number it is associated with. This number may indicate a relative brightness level (for in intensity level image) or a distance measure (for a range image. The analysis of two-dimensional signals, image processing, has come a long way in the last three decades. Several good books on image processing are currently available, for instance, [2, 21, 22, 34, 61, 62, 64]. Image processing has evolved into a very broad area and has spawned off a number of subfields such as image compression, image databases, image analysis etc. This dissertation deals with two specific subfields of image processing, namely edge detection and segmentation.

The study of edge detection and segmentation is useful since these fields form the basis for several other areas in image processing, and advances in these fields have a profound impact on image processing as a whole.

1.2.1 Segmentation

The process of separating an image into two or more subimages based on some uniformity (homogeneity) criterion is called segmentation. For instance, we may collect all pixels below a certain intensity level into one subimage and the complement of this subimage with respect to the original image may be a second subimage — a process called *thresholding*. Loosely speaking, segmentation is the dual problem of edge detection. While the general idea in edge detection is the identification of those (possibly few) pixels in an image which are *different* from those in their immediate neighborhood, the object of segmentation is, in some sense, the opposite; that of identifying those (possibly many) pixels which are *homogeneous* with those in their immediate neighborhood.

Several segmentation techniques have appeared in the literature in the last ten or so years. Haddon and Boyce [26] describe a segmentation technique that uses both boundary and region information. Pappas [52] describes an adaptive bottom-up algorithm for segmenting images that performs better than the K-means algorithm. Other segmentation methods proposed in the literature include [5, 9, 38, 40, 60, 83].

This dissertation proposes and evaluates two different but related algorithms ($S1$ and $S2$; see Chapter 4) for segmenting images. Both algorithms employ a multiresolution technique for analyzing the histogram of an image and proceeding from

an initial crude segmentation to increasingly better segmentations of the original image by a process of controlled refinement.

The basic idea behind both these segmentation algorithms is that of projecting the image histogram onto different subspaces of $L^2(\mathbf{R})$ — the space of square integrable functions of one variable. The initial (coarse) segmentation is performed by relying on the projection of the histogram onto a very coarse subspace of $L^2(\mathbf{R})$. If the segmentation obtained is not satisfactory, it is refined by projecting the histogram onto increasingly finer subspaces. These ideas will be made more precise in Chapter 4. The algorithms are linear in the size of the image, and are faster than split-and-merge type algorithms since they operate on the image histogram instead of the image. Further, each algorithm makes exactly two passes over the entire image.

1.2.2 Edge Detection

An *edge* is an intensity level discontinuity in an image. An edge may occur in an image for a variety of reasons; for example, the edge may signal the presence of an "object" in the image, or it may be a result of the shadow thrown by an object elsewhere in the image. The process of edge detection then, involves the location and isolation of significant intensity variations in images. Recently, there has been a lot of interest in detecting edges in images at various scales, the so-called multiscale edges.

The most common method of detecting edges uses differential operators of various types. Marr and Hildreth [47] proposed the use of the second derivative operator for locating edges. The zero-crossings of the second derivative of the image correspond to sudden changes in intensity in the image. Canny [8] describes an edge

detection scheme that uses appropriately scaled Gaussian functions to isolate edges from modulus extrema of the gradient of the image convolved with the Gaussian functions.

The main problem with a derivative based approach to edge detection is that numerical differentiation of images, fundamental to the edge detection process, is ill-defined in the Hadamard sense; for details see [72]. Thus, an image needs to be regularized before the differentiation is carried out. There are a number of choices for regularizing filters all of which correspond to some sort of a smoothing operation performed on the image. Nalwa and Binford [50] describe an edge detection technique that fits surfaces to subimages using the least-squares criterion, detects short linear edge elements and approximates an edge by a piecewise linear curve composed of these edge elements.

This dissertation describes an edge detection scheme that uses a segmentation scheme (see next section) as a preprocessor. The primary motivation for this approach was the search for a technique that performed better than traditional techniques on images with weak gradients such as satellite oceanographic images [33].

1.3 Multiresolution and Wavelets

The main advantage of using wavelet-based techniques as against, say, Fourier-based methods is the fact that wavelet-based techniques achieve good localization properties in both the signal and the transform domain. The Fourier transform, while achieving good localization in the frequency domain, suffers from poor localization in the time domain. In fact, the Fourier transform is unable to provide any

time domain localization information at all! The other advantage of wavelet theory is the fact that it provides very general tools and techniques for signal processing and consequently has innumerable potential applications.

Multiresolution provides a natural framework for analyzing phenomena. It is believed that the human visual system incorporates a hierarchical scheme for extracting detail from natural scenes. For example, when we visit an unfamiliar area, we are first aware of large landmarks such as trees and buildings and as we grow more familiar with the surroundings, we recognize smaller features such as subtle variations in the shades of color of the roads and the foliage on the trees.

Wavelets and multiresolution appear in the literature on image processing for a variety of applications such as singularity detection [44], image coding using multiscale edges [46, 19], image coding using vector quantization on the transform coefficients [1], feature detection [37, 39, 70] and geometrical registration of images [16]. For an excellent description of signal decomposition using the idea of multiresolution, see [41, 42, 43]. Meyer [48, 49] provides a thorough and readable introduction to wavelets.

Wavelets are also understood, from a signal processing perspective, through the study of filter banks. For an excellent tutorial on the general theory of digital filters and filter banks see [74]. More information about multirate filter banks can be found in [75, 76, 77, 31, 32, 73]. Walter [79] gives a good introduction to wavelets in the context of other orthogonal systems. Strang [66] gives a readable, non-technical description of wavelets and applications. For a comparison of wavelet transforms and Fourier transforms, see Strang [67]. For a discussion of time-frequency localization in signal processing, see [15].

1.4 Motivation

This dissertation is motivated mainly by an ever-present need for more efficient techniques for analyzing, manipulating and understanding the visual world around us using computers. The problems of edge detection and segmentation have been chosen since they form the basic building blocks of any complex image processing system, and an understanding of the basics is essential for any success with understanding more complex systems.

Although the literature has seen an ever increasing number of applications for wavelets and multiresolution analysis, it is still not clear what advantages wavelets have to offer in real world image processing applications. For instance, at the time of this writing, we are yet to see any widely available commercial products, hardware or software, that actually implement algorithms using wavelets or multiresolution, with the possible exception of the pyramid architecture for image processing. This is a secondary motivating factor for the current dissertation.

The emphasis, here, is on the computational aspects of wavelets. The various algorithms and techniques proposed in this dissertation have been tested on a number of test images. These images have been drawn from different sources and include satellite oceanographic images, images of natural scenes etc. The modeling or analysis of texture in images is beyond the scope of this dissertation and as such, the test images used do not contain a lot of texture.

Chapter 2

Preliminaries

In this chapter, we briefly review the notation used in the dissertation and introduce the mathematical concepts which will be often used in the remainder of the dissertation. It is assumed that the reader is familiar with the fundamental concepts of analysis and the theory of Fourier transforms.

2.1 Notation

The sets of integers, non-negative integers, reals and non-negative reals are respectively denoted by $\mathbf{Z}$, $\mathbf{Z}_+$, $\mathbf{R}$ and $\mathbf{R}_+$.

$L^2(\mathbf{R})$ denotes the space of measurable square-integrable functions of one real argument, $f(x)$. For two functions $f,\ g \in L^2(\mathbf{R})$ the inner product of $f(x)$ with $g(x)$ is given by

$$\langle f(x),\ g(x)\rangle = \int_{-\infty}^{\infty} f(x)\ g(x)dx\ . \tag{2.1}$$

The norm of $f(x) \in L^2(\mathbf{R})$ is given by

$$\|f\|^2 = \langle f,\ f\rangle = \int_{-\infty}^{\infty} |f(x)|^2\ dx\ . \tag{2.2}$$

The convolution of two functions $f,\ g \in L^2(\mathbf{R})$ is denoted by

$$f * g(x) = \int_{-\infty}^{\infty} f(\tau)\ g(t-\tau)d\tau\ . \tag{2.3}$$

The *Fourier transform* of $f(x) \in L^2(\mathbf{R})$, written $\mathcal{F}f(x)$, is given by

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x)\ e^{-i\omega x}\ dx \tag{2.4}$$

and is denoted by $\hat{f}(\omega)$. For $f(x) \in L^2(\mathbf{R})$, the Fourier Transform $\hat{f}(\omega) \in L^2(\mathbf{R})$. The following properties of the Fourier Transform are easy to derive (a and τ are arbitrary scalars).

1. $\mathcal{F}f(ax) = \frac{1}{a}\ \hat{f}(\frac{\omega}{a})$
2. $\mathcal{F}f(x-\tau) = e^{i\tau\omega}\ \hat{f}(\omega)$
3. $\mathcal{F}f'(x) = i\omega\ \hat{f}(\omega)$
4. $\mathcal{F}xf(x) = \hat{f}'(\omega)$

An image is a two-dimensional array of numbers drawn from a finite set, and is of finite size. Thus, it may be considered to be in the function space $L^2(\mathbf{R}^2)$. The norm of $f(x,y) \in L^2(\mathbf{R}^2)$ is similar to the norm in the function space of one-dimensional functions and is given by

$$\|f\|^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^2\ dx\ . \tag{2.5}$$

Likewise, the Fourier transform of $f(x,y) \in L^2(\mathbf{R}^2)$ is given by

$$\hat{f}(\omega_x,\omega_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\ e^{-i(\omega_x x+\omega_y y)}\ dx\ dy \tag{2.6}$$

and is denoted by $\hat{f}(\omega_x,\omega_y)$. The Fourier transform of a function gives frequency information in a signal but this information is not well localized in time. For instance, to investigate the characteristics of a function at a particular instant in time, the whole gamut of frequencies from $-\infty$ to ∞ need to be considered.

The *Window Fourier Transform*

$$Gf(\omega,u) = \int_{-\infty}^{\infty} e^{-i\omega x}\ g(x-u)\ f(x)\ dx \tag{2.7}$$

measures the amplitude of the sinusoidal component of f with frequency ω around the point u in the time domain. By translating the *window function* g, the entire real line $\mathbf{R}$ can be covered. In other words, the set of functions

$$\left\{\int_{-\infty}^{\infty} e^{-i\omega x}\, g(x-u)\, f(x)\, dx\right\}_u \in \mathbf{R} \tag{2.8}$$

can be used to analyze the sinusoidal component of f with frequency ω anywhere in the time domain. When the Gaussian function

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.9}$$

is chosen as the window function, the window Fourier transform becomes the well-known *Gabor Transform*. Thus, the window Fourier transform is a generalization of the Gabor transform. In the phase-space representation, the window Fourier transform corresponds to a uniform sampling of both the time and frequency domains. Thus, at high frequencies, the *resolution* of the transform is not sufficient to discern fine variations in the signal. Further details about the window Fourier transform are found in [43]. Teuner and Hosticka [71] describe the use of an adaptive version of the Gabor transform for image processing.

A function $g(x)$ is called a *smoothing* function if it converges uniformly to 0 at $\pm\infty$ and

$$\int_{-\infty}^{\infty} g(x) = 1\ . \tag{2.10}$$

The *dilation* of a function $f(x) \in L^2(\mathbf{R})$ by a scale factor s is written

$$f_s = \frac{1}{s} f(\frac{x}{s}) \tag{2.11}$$

where, if $s > 1$, the operation is equivalent to "expanding" f, and if $s < 1$, it corresponds to "compressing" f. In general, $s \in \mathbf{R}$. When $s = 2^j$, f_s is called a

binary dilation of f is denoted f_{2^j}. From a computational perspective, it is efficient to study binary dilations of functions. For instance, Mallat and Zhong [46] have shown that signals are accurately characterized by studying the local extrema in the convolution product of a signal with a binary dilation of a smoothing function. The scale factor of the smoothing function determines the scale at which signal properties are studied.

2.2 Multiresolution Analysis

An important concept in wavelet analysis is the idea of a multiresolution analysis of function spaces. This idea provides a natural framework for understanding wavelets. Several excellent overviews of this framework exist in the literature. For example, Mallat [41, 42], and Jawerth and Sweldens [35]. A multiresolution analysis of $L^2(\mathbf{R})$ is a set of closed, nested subspaces $V_j;\ j \in \mathbf{Z}$ satisfying the following properties.

1. $\bigcap_{k\in\mathbf{Z}} V_i = \emptyset$
2. $\overline{\bigcup_{k\in\mathbf{Z}} V_k} = L^2(\mathbf{R})$; the bar over the union indicates the closure in $L^2(\mathbf{R})$,
3. $\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots$,
4. $f(x) \in V_i \Leftrightarrow f(2x) \in V_{i+1}$,
5. $f(x) \in V_i \Leftrightarrow f(x-k) \in V_i$ for $k \in \mathbf{Z}$
6. There exists a *scaling function* $\phi(x) \in V_0$ such that $\{\phi(x-k)|k \in \mathbf{Z}\}$ forms a *Riesz* basis of $L^2(\mathbf{R})$.

Denoting the orthogonal complement of V_j in V_{j+1} by W_j, it is easy to see that

$$\bigoplus_{k \in \mathbf{Z}} W_k = L^2(\mathbf{R}) . \tag{2.12}$$

Thus, if we have an orthonormal basis (say, $\{\psi\}_j$) for some W_j, we can generate an orthonormal basis for $L^2(\mathbf{R})$ by scaling and translating the basis for W_j.

In practice, we usually deal with discretely sampled functions which indicates the existence of an implied "finest" scale. Further, since we have a finite set of sample points for the function under study, there is also an implied "coarsest" scale J. Thus, we study sampled functions between two fixed scales, 1 and J, and the function being studied, resides in $V_1 \oplus \oplus_{k=1}^{J} W_k$.

2.3 Wavelets

In this subsection, we review the basic mathematical tools necessary for studying the properties of wavelets. For a more complete introduction and discussion, the interested reader is referred to [10, 12, 13, 14]. For more recent developments and applications, see [4, 7, 65, 78, 84].

Wavelets offer a hierarchical, orthogonal basis for analyzing functions. Though the theory of wavelets is relatively new, orthogonal bases for function analysis have been around for a while; for example, the Fourier basis, the Hermite functions and the earliest example of a multiresolutional, orthogonal function basis, the Haar system (see [25]).

2.3.1 Window Functions

A function $f(x) \in L^2(\mathbf{R})$ is called a window function if $xf(x) \in L^2(\mathbf{R})$. The *center* $\overline{x}$ of f is defined as

$$\overline{x} = \frac{1}{\|f\|^2} \int_{-\infty}^{\infty} x|f(x)|^2 dx$$

and the *radius* Δ_f of f is defined as

$$\Delta_f = \frac{1}{\|f\|} \left\{ \int_{-\infty}^{\infty} (x - \overline{x})^2 |f(x)|^2 dx \right\}^{\frac{1}{2}} .$$

The width of the window function is thus $2\Delta_f$. The center and radius of the window function are analogous to the *mean* and the *standard deviation* of a statistical distribution.

2.3.2 The Continuous Wavelet Transform

The continuous wavelet transform (CWT) of a function $f(x) \in L^2(\mathbf{R})$ is given by

$$\mathcal{W}f(a,b) = \langle f, \psi_{a,b} \rangle \; ; \quad \psi_{a,b} = \frac{1}{\sqrt{|a|}} \psi(\frac{x-b}{a}) \; a, b \in \mathbf{R}, a \neq 0 \tag{2.13}$$

The functions $\psi_{a,b}$ are scaled so that their L^2 norms are independent of a. Using Parseval's identity, we have

$$2\pi \mathcal{W}f(a,b) = \langle \hat{f}, \hat{\psi}_{a,b} \rangle \; ; \quad \hat{\psi}_{a,b} = \frac{a}{\sqrt{|a|}} e^{-i\omega b} \hat{\psi}(a\omega) \; . \tag{2.14}$$

This last equation follows directly from the properties of the Fourier transform (see equation 2.4).

If $\psi(x)$ and its Fourier Transform are both window functions with centers and radii given by

$$\begin{aligned}
\overline{x} &= \frac{1}{\|\psi\|^2} \int_{-\infty}^{\infty} x|\psi(x)|^2 dx \; , \\
\Delta_x &= \frac{1}{\|\psi\|} \left\{ \int_{-\infty}^{\infty} (x - \overline{x})^2 |\psi(x)|^2 dx \right\}^{\frac{1}{2}} \; , \\
\overline{\omega} &= \frac{1}{\|\hat{\psi}\|^2} \int_{-\infty}^{\infty} \omega |\hat{\psi}(\omega)|^2 d\omega \; ,
\end{aligned}$$

$$\Delta_\omega = \frac{1}{\|\hat{\psi}\|}\left\{\int_{-\infty}^{\infty}(\omega-\overline{\omega})^2|\hat{\psi}(\omega)|^2 d\omega\right\}^{\frac{1}{2}}, \tag{2.15}$$

the CWT localizes f, simultaneously, within a time interval $[b + a\overline{x} - a\Delta_x, b + a\overline{x} + a\Delta_x]$ and a frequency interval $[\frac{\overline{\omega}-\Delta_\omega}{a}, \frac{\overline{\omega}+\Delta_\omega}{a}]$. In the time-frequency plane, this localization corresponds to a rectangular *box* whose area is constrained, by Heisenberg's uncertainty principle, to be at least as large as 2. Thus, $4\Delta_x\Delta_\omega \geq 2$.

If ψ satisfies the so-called *admissibility condition*,

$$C_\psi = \int_{-\infty}^{\infty}\frac{|\hat{\psi}(\omega)|^2}{\omega}d\omega < \infty, \tag{2.16}$$

the CWT is invertible and f can be reconstructed as

$$f(x) = \frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mathcal{W}(a,b)\psi_{a,b}(x)\frac{dadb}{a^2}. \tag{2.17}$$

From the admissibility condition, it follows that $\hat{\psi}(0) = 0$ and we see why the name "wavelet" is appropriate for describing $\psi(x)$. A more complete discussion of wavelet transforms and their properties may be found in [23, 30, 58, 67, 80, 81].

2.3.3 The Dyadic Discrete Wavelet Transform

In the continuous wavelet transform (Eq. 2.13), if a is restricted to powers of 2, and b to integer multiples of a, we have the dyadic discrete wavelet transform (DWT)

$$\mathcal{W}f(j,k) = 2^{-j/2}\int_{-\infty}^{\infty} f(x)\ \psi(2^{-j}x - k)\ j,k \in \mathbf{Z}. \tag{2.18}$$

Denoting $2^{-j/2}\psi(2^{-j}x - k)$ by $\psi_{j,k}$, we may rewrite the above equation as

$$\mathcal{W}f(j,k) = (f, \psi_{j,k}). \tag{2.19}$$

Thus, the "wavelet coefficient" of f at scale j and translation k is the inner product of f with the appropriate basis vector at scale j and translation k.

The dyadic discrete wavelet transform, which refers to the collection of wavelet (and scaling function) coefficients forms a complete signal representation. In other words, the transform is invertible and yields the original signal without any error. The transform domain coefficients can also be used for frequency domain filtering of signals e.g. for low-pass and band-pass filtering of an image.

2.3.4 The Matrix View of the DWT

In practice, we always deal with discretely sampled signals (functions). In such cases, it is useful to understand the wavelet transform as a matrix-vector multiplication. Thus, if $f_n; \quad n \in [0, N-1]$ is a discretely sampled function, the discrete wavelet transform of f_n may be written as

$$\mathbf{W}\,\mathbf{f} = \hat{\mathbf{f}} \tag{2.20}$$

where $\mathbf{f}$ is the vector representing the sequence f_n and $\mathbf{W}$ is a band-diagonal matrix derived from the wavelet filter coefficients (for a more complete analysis, see [10, 12, 13, 55, 59]), and has the form shown in Figure 2.3.4. This form for the wavelet matrix corresponds to the 4-coefficient Daubechies wavelet filter. For longer filters, the structure is similar. The structure $\{c_0\ c_1\ c_2\ c_3\}$ is called a "smoothing" filter since it computes a weighted *sum* over four samples in the vector f_n. The structure $\{c_3\ -c_2\ c_1\ -c_0\}$ is called a "detail" filter since it computes a weighted *difference* over four samples in f_n.

From the structure of the matrix, it is clear that a vector of length N, upon being multiplied by $\mathbf{W}$, separates into two interlaced vectors, each of length $N/2$. The odd numbered rows give rise to "sums" and the even numbered rows give rise to "differences" in f_n. More precisely, the product of $\mathbf{W}$ and f_n is a column vector

$$\mathbf{W} = \begin{bmatrix}
c_0 & c_1 & c_2 & c_3 & & & & & & & & \\
c_3 & -c_2 & c_1 & -c_0 & & & & & & & & \\
 & & c_0 & c_1 & c_2 & c_3 & & & & & & \\
 & & c_3 & -c_2 & c_1 & -c_0 & & & & & & \\
\vdots & \vdots & & & & & \ddots & & & & & \\
 & & & & & & & c_0 & c_1 & c_2 & c_3 & \\
 & & & & & & & c_3 & -c_2 & c_1 & -c_0 & \\
c_2 & c_3 & & & & & & & & c_0 & c_1 & \\
c_1 & -c_0 & & & & & & & & c_3 & -c_2 &
\end{bmatrix}.$$

Figure 2.1: The wavelet transform matrix

that has the following form:

$$[\ s_0^1\ \ d_0^1\ \ s_1^1\ \ d_1^1\ \ \cdots\ \ s_{N/2}^1\ \ d_{N/2}^1\ ]^T\ .$$

The s^1's in the vector above correspond to the results of the computations of the smoothing filter and the d^1's correspond to those of the detail filter. In the language of multiresolution analysis, the s^1's represent the projection of f_n onto a space (say) V_0 and the d^1's, the projection of f_n onto the space W_0. Thus, the d^1's represent the wavelet coefficients at the finest scale. Now, $\mathbf{W}$ can be used again to project s^1's onto the second level sum and difference subspaces, V_{-1} and W_{-1}. The projection operation thus extracts wavelet coefficients in increasingly coarse scales until finally we are left with $s_0^{\log N}$ and $d_0^{\log N}$ which represent the scaling function coefficient and the wavelet coefficient at the coarsest scale. That this procedure ends, follows directly from the finiteness of the sampled function f_n.

Now, the original function f_n may be reconstructed exactly by inverting the matrix multiplication process. We start at the coarse scale and multiply the transform coefficient vector with $\mathbf{W^{-1}}$ and reconstruct a longer and longer sequence of s's each time. At the coarsest scale, one s and one d combine to give two s's at a finer scale. At this finer scale, two s's and two d's combine to give four s's and so on. The c_i's are chosen so that $\mathbf{W}$ is a unitary matrix and $\mathbf{W^T} = \mathbf{W^{-1}}$. Further details about the actual calculations that yield c_i's may be found in [55].

In this dissertation we concern ourselves primarily with the Daubechies wavelets which belong to the class of orthogonal wavelets. In signal processing parlance, these correspond to the class of filters with a maximally flat frequency response. These wavelets also maximize the smoothness of the scaling function and the wavelet function for a fixed filter length (number of filter coefficients). For a more thorough analysis of the construction and other properties of these wavelets, the interested reader is refered to [12, 13, 14]. The Daubechies filters belong to the larger class of what are called *perfect reconstruction filters*. The wavelets associated with these filters are also called *compactly supported* wavelets since the scaling function ϕ and the wavelet function ψ corresponding to these filters are nonzero only over a finite portion of the real line.

Chapter 3

The Discrete Wavelet Transform

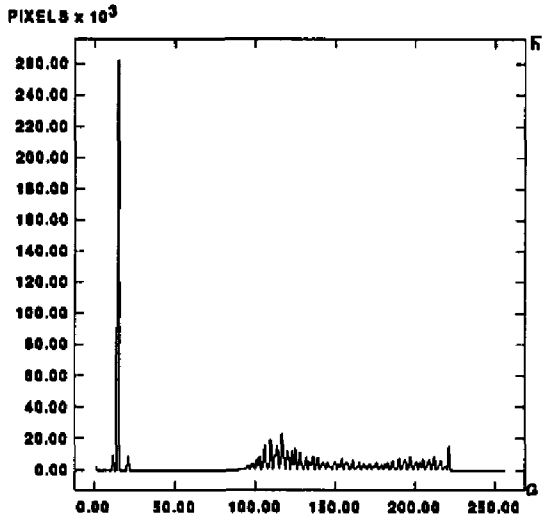
This chapter describes the discrete wavelet transform in some detail since this transform forms the basis for various algorithms used elsewhere in this dissertation. For example, in the wavelet-based smoothing of an image histogram (see Section 3.2), the coefficients of the one-dimensional wavelet transform are used to obtain a smoothed (low-pass) version of the histogram.

3.1 The One Dimensional Transform

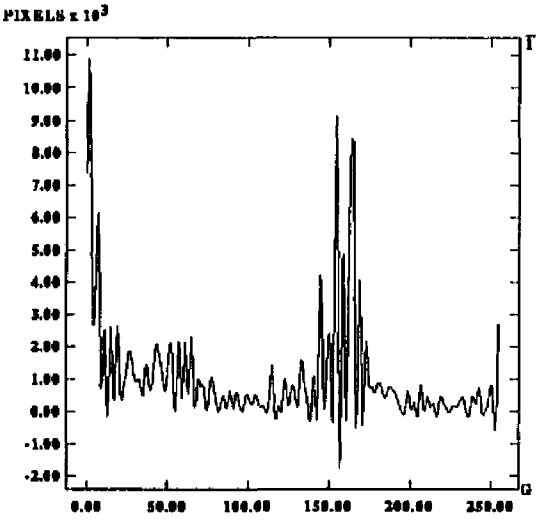
As described in Chapter 2, the discrete wavelet transform is a complete representation of a signal. Further, it provides important information about the time frequency localization of energy in a signal. The discrete wavelet transform of a signal is a collection of projections of the signal onto a set of wavelet subspaces W_j and a scaling function subspace V_0 as described in Section 2.3.3. The wavelet subspaces capture the fluctuations in the signal that the associated scaling function spaces cannot.

The discrete one dimensional wavelet transform is obtained by multiplying a signal vector with the matrix in Figure 2.3.4. Each time the difference coefficients are retained, and the sum coefficients are further multiplied by the same matrix. This process is continued until a single sum coefficient remains.

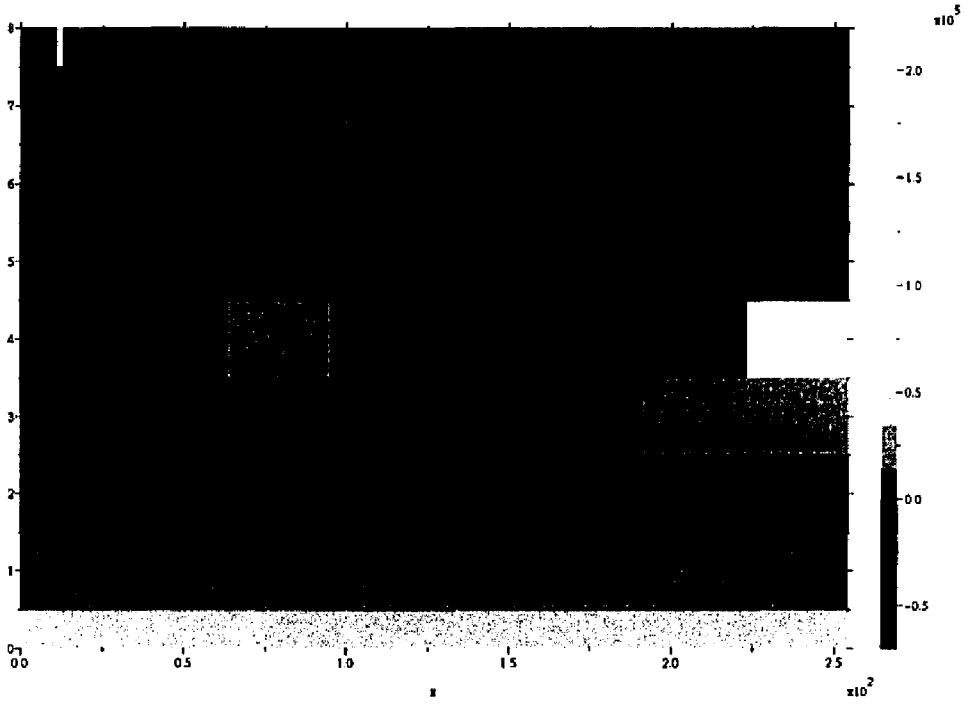
Figure 3.4(a) shows the histogram of a satellite image, and (b) shows the wavelet transform of the histogram computed using the Daubechies 4-coefficient



(a) histogram



(b) wavelet transform



(c) time-frequency diagram

Figure 3.1: Image histogram and its discrete wavelet transform

wavelet filter. Note that the transform coefficients have both positive and negative values though the original histogram has nonnegative values.

The wavelet transform is best visualized as a two-dimensional object, the *time-frequency* plot, which provides information about the location and magnitude of the wavelet coefficients. Figure 3.4(c) shows the time-frequency diagram for the wavelet transform of the histogram. Note the presence of coefficients of high magnitude at the top of the diagram corresponding to the large peak at gray level 15 and in the range 100 ... 220. The figure also shows a vertical strip of gray to the right of the time-frequency diagram indicating the correspondence between the shade of gray and the magnitude of the wavelet coefficient.

3.2 Histogram Smoothing

The coefficients of the discrete wavelet transform can be modified in several ways before the inverse wavelet transform reconstructs the original function. The overall effect is similar to frequency domain filtering. For example, setting the wavelet coefficients in the finest subspace to zero is equivalent to low-pass filtering the function.

Let h be a histogram of an image. Let $\hat{h}$ denote the discrete wavelet transform of h. By suppressing a subset of the wavelet coefficients of h in $\hat{h}$, it is possible to reconstruct a frequency filtered copy of the original histogram. For instance, if all coefficients at the finest scale in the wavelet transform are set to zero before the inverse transform, the reconstructed histogram represents a low-pass version of the original with a passband that is one half the bandwidth of the original. By periodically extending the histogram on both sides of its support, it is possible to

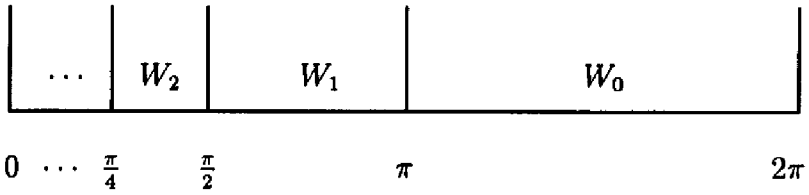


Figure 3.2: Frequency domain picture of wavelet spaces W_j

associate the range $[0, 2\pi]$ with the frequency bandwidth of h. Figure 4.5 shows the correspondence between the wavelet spaces W_j and the frequency bands within the $[0, 2\pi]$ range.

Figures 3.3 and 3.4 show the effect of smoothing a histogram by eliminating wavelet coefficients from its discrete wavelet transform. In each figure, (a) shows the original histogram, (b) shows the same histogram low-pass filtered by eliminating one level of wavelet coefficients, (c) shows the effect of eliminating four levels of coefficients and (d) shows the effect of eliminating seven levels of coefficients. It is interesting to note that although (c) and (d) look significantly different from the original histogram in (a), the locations of maxima and minima in the filtered histogram do have a strong correlation with the maxima and minima in the original histogram. Thus, the wavelet smoothing operation remains faithful to the original function (in the L^2 sense) at all levels of smoothing.

As more and more coefficients in the fine subspaces W_j are set to zero, the frequency passband of the histogram reduces. For better frequency resolution, it is possible to filter the histogram using wavelet coefficients of functions which span the upper half of the frequency band. For example, if W_0 is expressed as the nonoverlapping union of subspaces U_j, the projection of the original histogram onto

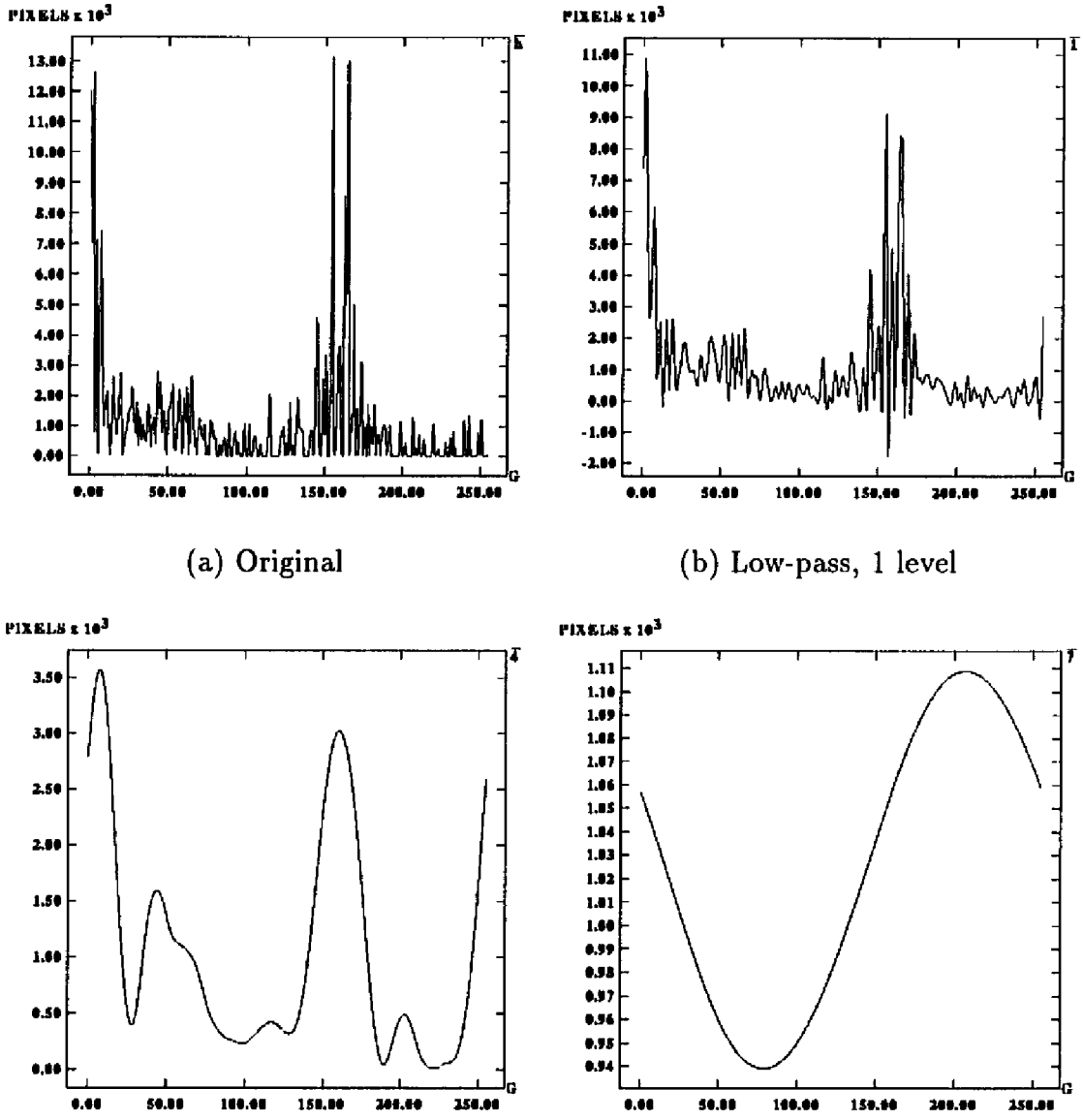
these latter subspaces yields wavelet coefficients of the wavelets which span U_j which can be used for a more precise frequency domain filtering of the histogram. This operation is the direct analogue of m-channel filtering of signals in filter bank theory.

An alternative to the m-band wavelet idea is that of further subdividing the wavelet spaces into smooth and detail spaces with respect to some other wavelet basis. In other words, select spaces V_0' and W_0' such that $W_0 = V_0' \oplus W_0'$ and obtain orthogonal bases for V_0' and W_0' to partition W_0 better. A generalization of this idea leads to the matching pursuits algorithm of Mallat and Zhang [45].

It is important to notice that setting a wavelet coefficient of a histogram to zero and reconstructing the histogram does not change the area under the histogram. This follows directly from the fact that the area under a wavelet function is zero, and thus the wavelet coefficient contributes nothing to the area under the histogram. A nonlinear filtering operation such as median filtering, for example, changes the area under the histogram. Since the area under the histogram corresponds to the pixel count in an image, the filtered histogram is, in some sense, no longer faithful to the image. The idea of low-pass filtering an image histogram using its wavelet transform can also be extended to two dimensions as will be seen in section 3.4.

3.3 The Two Dimensional Transform

A multiresolution analysis of $L^2(\mathbf{R}^2)$ may be obtained by a tensor product of two multiresolution analyses of $L^2(\mathbf{R})$. This enables the computation of the wavelet transform of a two-dimensional signal, an image, as a separable transform — a row/column transform followed by a column/row transform. While in the one dimensional case we had one scaling function subspace and one wavelet subspace



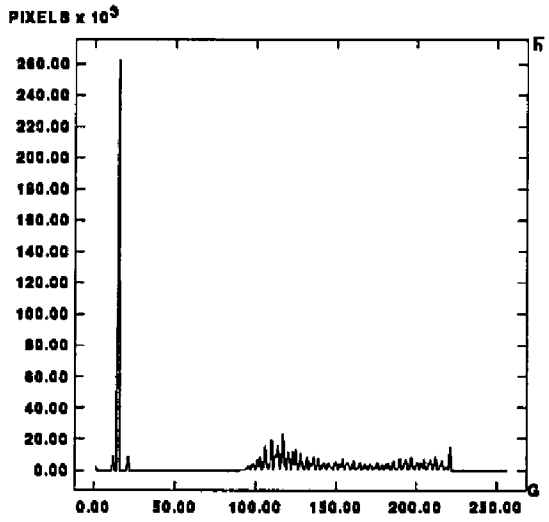
(a) Original

(b) Low-pass, 1 level

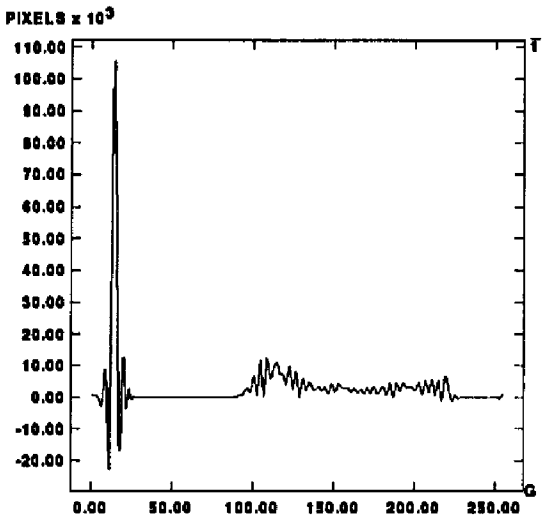
(c) Low-pass, 4 levels

(d) Low-pass, 7 levels

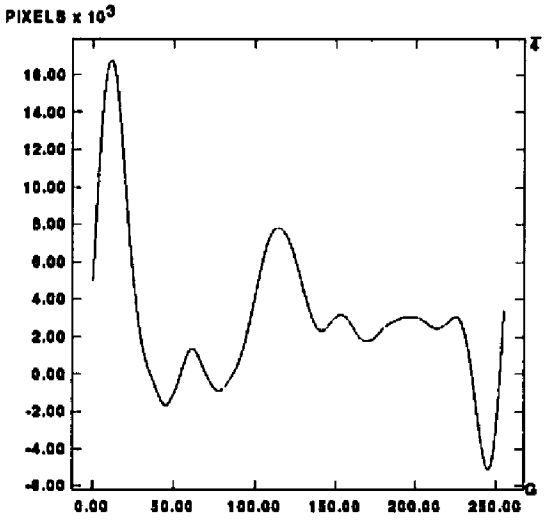
Figure 3.3: Wavelet smoothing of an image histogram



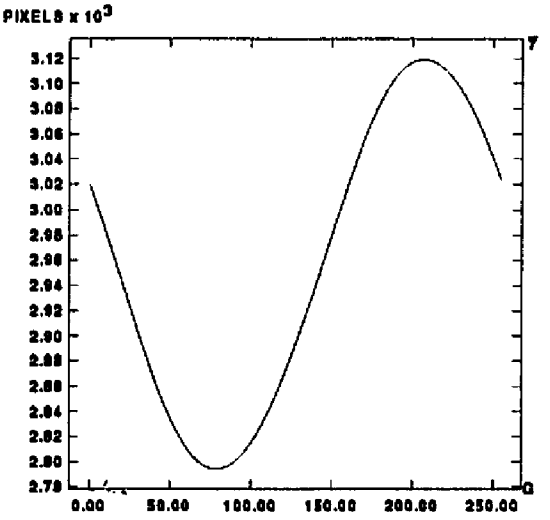
(a) Original



(b) Low-pass, 1 level



(c) Low-pass, 4 levels



(d) Low-pass, 7 levels

Figure 3.4: Wavelet smoothing of an image histogram

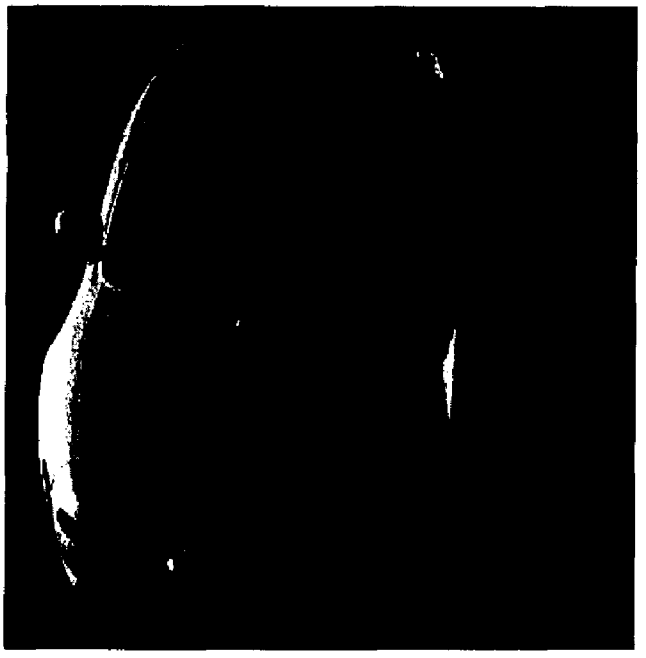
at each scale, the two-dimensional case leads to one scaling function subspace and three wavelet subspaces at each scale. This is best understood as follows. In the one dimensional case, $V_0 \oplus W_0 = V_1$. In the two-dimensional case, we have $(V_0 \oplus W_0) \otimes (V_0 \oplus W_0) = V_1$ where $\otimes$ represents the tensor product. Thus,

$$V_1 = V_0 \otimes V_0 \bigoplus V_0 \otimes W_0 \bigoplus W_0 \otimes V_0 \bigoplus W_0 \otimes W_0 \tag{3.1}$$

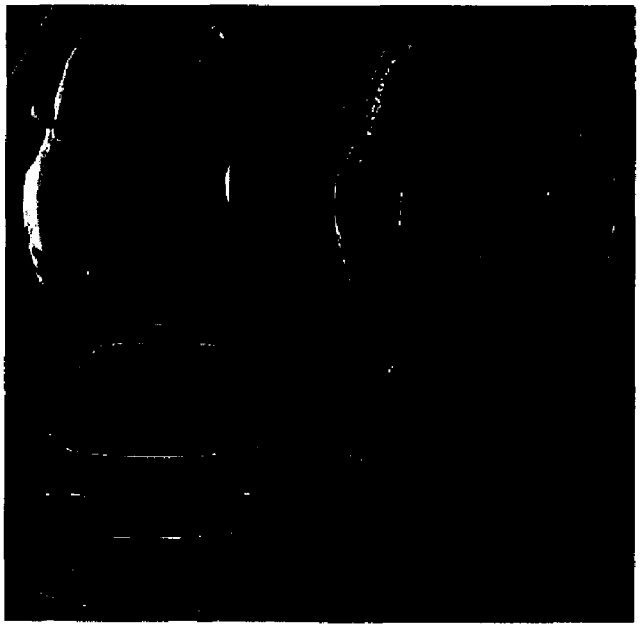
where $V_0 \otimes V_0$ represents the smooth subspace and the other three tensor products represent the detail or the wavelet subspaces.

The discrete wavelet transform of an image then, is the projection of the image intensity function (which is finitely supported) onto the V's and the W's associated with the choice of a wavelet. From (Eq. 3.1), the projection of an image onto V_1 may be obtained as the direct sum (denoted by $\oplus$) of the four projections onto the subspaces on the right hand side. Figure 3.5 shows an image and its projections. The wavelet filter used to compute this transform is the well-known Daubechies 4 coefficient filter [13]. Now, the low-resolution image, which is the projection of the image onto $V_0 \otimes V_0$, can itself be expressed as the direct sum of four subspaces obtained by writing V_0 as $V_{-1} \oplus W_{-1}$ and expanding the tensor product $V_0 \otimes V_0$.

This process of projecting the image onto successively coarser spaces cannot go on indefinitely since the image is of finite size. For purposes of image coding, the multiresolution process may be stopped wherever desired. In Figure 3.5(b), the detail image in the northeast corner, represents the difference information in the row transform and is thus sensitive to vertical edges. Similarly, the detail image in the southwest corner is sensitive to horizontal edges and the image in the southeast corner, to edges oriented at 45 degrees to the vertical.



(a) image



(b) wavelet transform (1 level)

Figure 3.5: An image and its wavelet transform

3.4 Image Smoothing

The dyadic wavelet transform described in the preceding discussion is a complete signal representation whenever the basic wavelet chosen satisfies certain conditions. The dyadic wavelet transform of a finite-length signal is itself a finite-length sequence. The transform represents the information contained in signals in a non-redundant fashion. The main advantage of using the dyadic wavelet transform instead of the more conventional Fourier transform is its spatial localization property. These general remarks will soon become clear in the following sections.

We now turn our attention to the problem of signal filtering in scale space, or what is the same thing, frequency-domain filtering. Briefly, the problem is as follows. Given a spatial (or time-domain) signal, how can certain frequencies in the transform domain be suppressed ? For example, *low-pass* and *high-pass* filtering operations remove high and low frequency components from the spectrum of a signal. In this chapter, we describe these filtering operations in connection with the dyadic wavelet transform. A high-pass filter may be used to suppress signal effects which have a high spatial frequency. Thus, noise in signals may be removed by suppressing the frequencies in the transform domain where it is likely to occur.

The wavelet transform of a signal can be used to filter the frequency components in the signal. For instance, frequencies in a specific band may be removed or retained during the filtering operation, by appropriately modifying the wavelet coefficients.

An image f can be considered to be an element of $L^2(\mathbf{R}^2)$. A multiresolution of $L^2(\mathbf{R}^2)$ can be obtained from a tensor product of spaces in $L^2(\mathbf{R})$. For example, consider a continuous function g in $L^2(\mathbf{R})$ discretely sampled to yield $g_i; i \in Z$. g_n may be thought of as existing at some finest scale n; in other words, g_i lies in

some subspace V_n of $L^2(\mathbf{R})$, and has no components in $L^2(\mathbf{R})$- V_n. Thus, it may be thought of as the projection of g onto V_n. In a similar manner, an image f may be considered to be the projection of some continuous function in $L^2(\mathbf{R}^2)$ onto some fine subspace $V_n \otimes V_n$ of $L^2(\mathbf{R}^2)$. Recalling from the multiresolution analysis of $L^2(\mathbf{R})$ that $V_n = V_{n-1} \oplus W_{n-1}$, we can write

$$V_n \otimes V_n = V_0 \otimes V_0 \oplus \bigoplus_{j=0}^{n-1} [V_j \otimes W_j \oplus W_j \otimes V_j \oplus W_j \otimes W_j] \tag{3.2}$$

where the $\otimes$ operator takes precedence over the $\oplus$ operator. If we associate ϕ_j, the scaling function and ψ_j, the wavelet (at scale j) with the subspaces V_j and W_j respectively, any f in $L^2(\mathbf{R}^2)$ may be written as

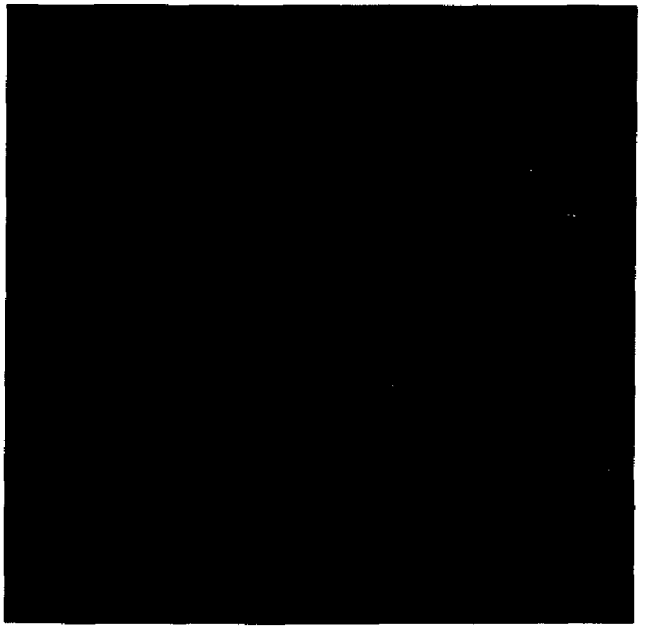
$$\begin{aligned} f &= \sum_k \sum_l a_{k,l}\, \phi_k \phi_l \\ &+ \sum_{j=0}^{n-1} \sum_k \sum_l [b^{\phi\psi}_{j,k,l} \phi_{j,k} \psi_{j,l} + b^{\psi\phi}_{j,k,l} \psi_{j,k} \phi_{j,l} + b^{\psi\psi}_{j,k,l} \psi_{j,k} \psi_{j,l}] \end{aligned} \tag{3.3}$$

where k and l depend on the support of f in the spatial domain.

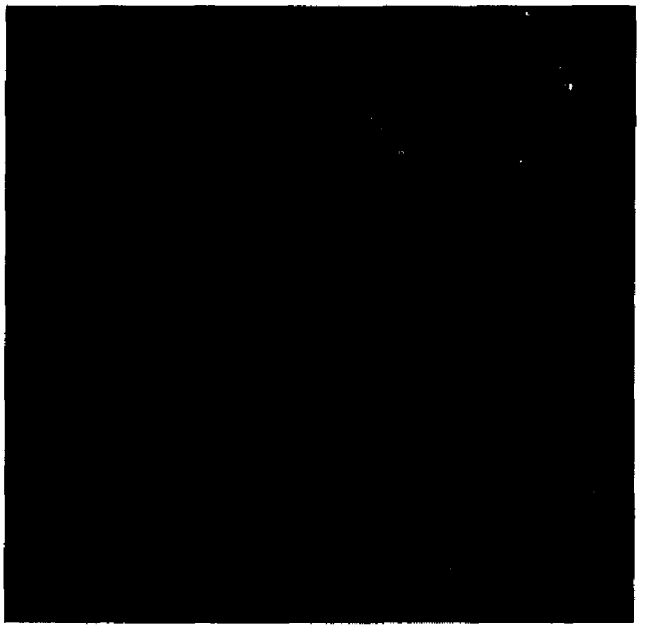
Setting a subset of the wavelet coefficients $b_{j,k,l}$ to zero in the above equation, we may obtain $\tilde{f}$ which is a smoothed version of f. For example, setting all b's at $j = n-1$ to zero in the equation yields a smoothed version of f which now resides in $V_{n-1} \otimes V_{n-1}$. The projections of f onto the subspaces $V_{n-1} \otimes W_{n-1}$, $W_{n-1} \otimes V_{n-1}$ and $W_{n-1} \otimes W_{n-1}$ have been filtered out leaving a coarser approximation of f. We will later (in Chapter 5) employ this technique for extracting edges in images while simultaneously filtering out high frequency noise which leads to spurious edges.

Figure 3.6 (a) and (b) show an image smoothed using this idea. In (a), only the finest resolution wavelet coefficients have been set to zero, while in (b), all coefficients in the three finest resolution wavelet spaces have been set to zero.

The loss of high frequency detail due to smoothing is clear when Figure 3.6 is compared with the original image in Figure 4.8 (a). A common image compression



(a) 1 level



(b) 3 levels

Figure 3.6: Wavelet smoothed image

technique is the use of standard coding algorithms to efficiently code the information in the wavelet subspaces, that is, the wavelet coefficients in the wavelet spaces. By discarding coefficients with small absolute values, a highly economical representation of the original image can be obtained. Further, a natural tradeoff exists between the compression ratio and the quality of the decompressed image, and this tradeoff is easily quantified in terms of the fraction of the wavelets retained in the encoding of the image.

Chapter 4

Segmentation

This chapter describes a multiresolution decomposition scheme for the histogram of an image. Information obtained from this decomposition is used for efficient, hierarchical segmentation of the image. Two different but related algorithms are proposed for the segmentation problem.

Algorithm *S1* uses conventional smoothing operations on the histogram of an image to smooth out rapid variations and employs a dyadic multiresolution decomposition scheme for extracting the peaks and valleys in the smoothed histogram. These features in the histogram are used to select threshold intensity levels for separating the image into subimages.

Algorithm *S2* uses the wavelet-based smoothing technique described in section 3.4.1 and obtains a set of projections of the original histogram onto a set of nested subspaces of $L^2(\mathbf{R})$. The projected histograms are arranged according to the coarseness of the underlying subspace. Then, starting with the coarsest subspace, the modality of the projected histogram is analyzed for a possible selection of coarse thresholds for a coarse segmentation of the image. A refinement of this initial segmentation is possible by examining the projected histograms in increasingly finer subspaces.

4.1 Introduction

Segmentation is an important low-level image processing task[61]. The goal of segmentation is the grouping of pixels in an image according to some arbitrary measure (criterion). For instance, all pixels in an image whose intensity levels lie within a specified range may be grouped together as one region. Pixels grouped together are considered to be *homogeneous* according to the criterion used in the segmentation process. The simplest type of image segmentation is based on the statistical distribution of intensity levels in the image and involves the selection of a *threshold* from the image histogram. The image is subsequently thresholded, resulting in a labeling of pixels according to whether their intensity level is greater than or less than the threshold. This method depends on the bimodal property of the histogram [54]. Often, histograms do not display bimodal properties. Only relatively noise-free images with a well defined background and a well defined foreground, have bimodal histograms. This limits the utility of the segmentation method just described.

This chapter deals with a multiresolution technique for analyzing histogram data in images. Thus, multiresolution is not applied directly to the image data but to the image histogram. The most significant advantage of this approach is the reduced computational effort resulting from processing the histogram instead of the entire image, at several levels of resolution. The number of intensity levels in an image (typically a power of 2) is usually much smaller than the size of the image in pixels. However, it must be remembered that at least one pass over the entire image is required to arrive at the histogram. The other advantage of applying multiresolution decomposition to the histogram is that the multiresolution process is applied in one dimension only; multiresolution of image data involves

two-dimensional analysis and decomposition. One dimensional multiresolution is simpler both in concept and in implementation.

The histogram is analyzed, first at a coarse scale and later at increasingly finer scales, for modality information. The resolution process is stopped when an acceptable segmentation of the image is obtained. Alternatively, the resolution process can be stopped when the image has been segmented into a specified number of regions. This chapter describes an algorithm for analyzing multimodal histograms and segmenting images hierarchically. The coarse segmentation of the image obtained by analyzing the histogram at a coarse scale is refined at higher levels of resolution.

The motivation for this work stems from a multiresolution architecture for segmenting images with bimodal histograms developed by Bongiovanni et al. [5]. Their approach involves a pyramid architecture for determining the bimodality of image subregions which are later integrated to arrive at the segmented image.

The quantitative analysis of the performance of various algorithms for segmentation has remained a fuzzy area. There are no well-defined metrics for evaluating the results of a segmentation. The main reason for this is the fact that image segmentation is an ill-defined problem. It is the grouping of pixels in an image according to some arbitrary measure. This implies that a set of pixels perceived to be homogeneous according to this arbitrary measure are classified as a single group and no other pixels in the image, outside this set, satisfy this homogeneity condition. Thus, different conditions for homogeneity lead to different segmentations of the same image, each segmentation as valid as any other.

4.2 Multiresolution Decomposition

For every $j \in \mathbf{Z}$ let

$$d_j = \{\frac{k}{2^j}; k \in \mathbf{Z}\} \tag{4.1}$$

denote the *dyadic rationals* at resolution level j. Thus, for any $j \in \mathbf{Z}$, d_j is a set of equally spaced sampling points on the real line $\mathbf{R}$. If $i < j$, d_i represents a *coarser* (lower resolution) sampling of $\mathbf{R}$, and if $i > j$, d_i represents a *finer* (higher resolution) sampling of $\mathbf{R}$. Note that $\ldots \subset d_{j-1} \subset d_j \subset d_{j+1} \ldots$ and the closure of $\bigcup_{j=-\infty}^{\infty} d_j$ is dense in $\mathbf{R}$.

Let $f : \mathbf{Z} \times \mathbf{Z} \rightarrow [0, g] \cap \mathbf{Z}$ be an image; g is the largest intensity level occurring in the image. Let $h_f : [0, g] \cap \mathbf{Z} \rightarrow \mathbf{Z}_+$, where $\mathbf{Z}_+$ is the set of all nonnegative integers, denote the histogram of the image. In other words,

$$h_f(k) = |\{(x, y) : f(x, y) = k\}|; k \in [0, g] , \tag{4.2}$$

where $|\cdots|$ denotes the cardinality operator on sets. Note that h_f is a discrete function. We construct a continuous function from h_f as described below and through a convenient abuse of notation, henceforth refer to this continuous function as the histogram function h_f. We let $h_f(x) = h_f(k)$ for $x \in [k, k+1)$. Thus, h_f is the union of several piecewise constant functions. Now, the domain of h_f is $\mathbf{R}$ and further, h_f has the property of compact support.

For $j \in \mathbf{Z}$, $h_f(x)$ can be sampled at sample points $\{d_j\}$ to yield a representation of h_f, at resolution level j, denoted h_f^j. Further, this representation can be expressed as a linear combination of a set of functions obtained by scaling and translating the characteristic function of the unit interval $[0, 1]$

$$\phi(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} . \tag{4.3}$$

ϕ is the well-known Haar scaling function. Consider the set of functions

$$\{\phi(2^j x - n)\}_{n=-\infty}^{\infty} \quad ; \tag{4.4}$$

each function in this set has as its support $[\frac{n}{2^j}, \frac{n+1}{2^j})$. Note that $[\frac{n_k}{2^j}, \frac{n_k+1}{2^j}) \cap [\frac{n_l}{2^j}, \frac{n_l+1}{2^j}) = \emptyset \ \ \forall \ \ k \neq l$, and $\cup_{n=-\infty}^{\infty} [\frac{n}{2^j}, \frac{n+1}{2^j}) = \mathbf{R}$. The relationship between the dyadic rationals and the intervals of support of ϕ is worth noting. From equation (4.4), a representation of $h_f(x)$ at the jth level of resolution is

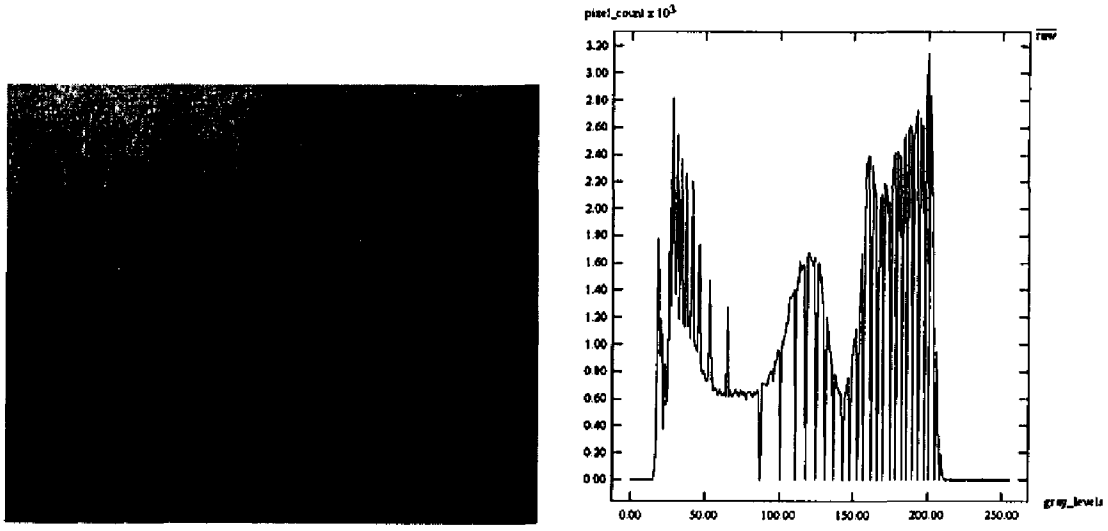
$$h_f^j(x) = \sum_{n=-\infty}^{\infty} h_f(2^{-j}n)\ \phi(2^j x - n) \tag{4.5}$$

where the summation is really over finitely many n since h_f has compact support. A representation of h_f at the highest resolution possible is obtained by sampling h_f at $\{d_0\}$.

4.3 Histogram Multiresolution

Figure 4.1 shows a typical image and the associated histogram of intensity levels occurring in the image. The histogram does not display well-defined modal properties. Multiresolution decomposition of this histogram involves sampling the histogram at several levels of resolution, starting from a coarse level and moving towards increasingly finer levels. Note that, although the histogram does not show well-defined modality, the envelope of the histogram does show three broad modes.

Before the histogram is analyzed at different resolution levels, it is low pass filtered to smooth the data. Neighborhood averaging, and morphological filtering are two satisfactory methods of data smoothing. After smoothing, the histogram is sampled at different resolution levels. This process is best understood as a sequence of projections onto successively finer subspaces of $L^2(\mathbf{R})$.



(a) (b)

Figure 4.1: A typical image and its histogram

Let $\phi(x)$ denote the Haar scaling function (eq. 4.3). Let

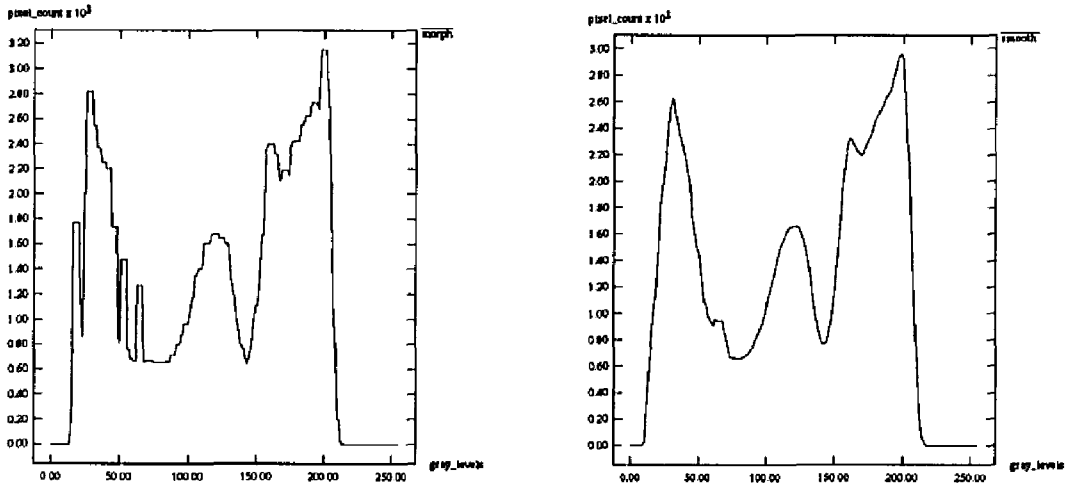
$$\phi_{j,k} = 2^{j/2}\,\phi(2^j x - k) \tag{4.6}$$

denote the scaling function ϕ at the scale factor j and at a translation of k. Let V_j be the spaces spanned by $\{\phi_{j,k}\}_{j,k\in Z}$, i.e. any function f in V_j can be written as

$$f = \sum_{k=-\infty}^{\infty} a_k\,\phi_{j,k};\ \sum_{k=-\infty}^{\infty} \mid a_k \mid^2 < \infty \tag{4.7}$$

where $a_k = \langle f, \phi_{j,k}\rangle$. Thus, the sequence $\{a_k\}_j$ represents the projection coefficients of f onto V_j.

The local extrema (peaks and valleys) of f , as manifest in the projection of f on V_j can be used to analyze f approximately. For instance, if f is the histogram of an image, various approximations may be obtained for f by projecting it onto the spaces V_j. The modes of f , its local maxima and minima, are faithfully represented in the sequences $\{a_k\}_j$ for various j. Thus, threshold values may be selected for



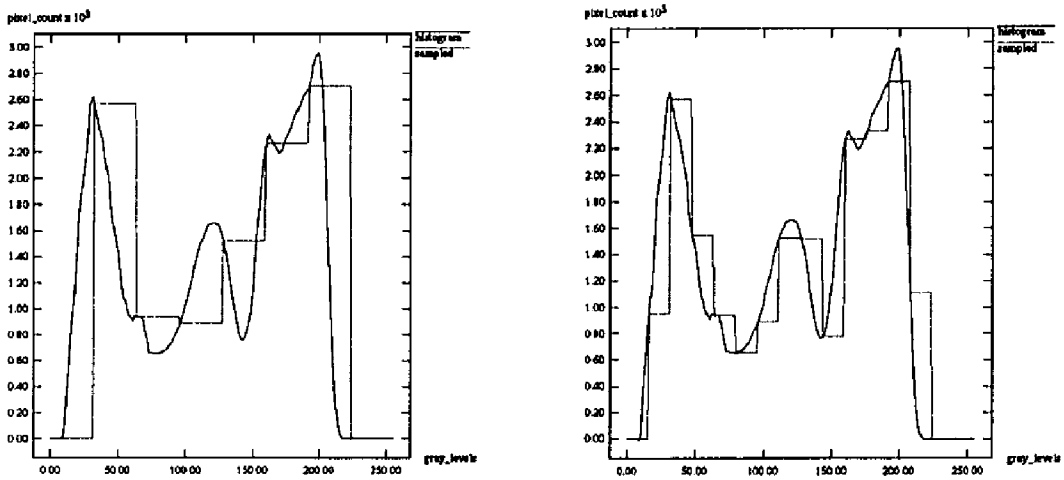
(a) (b)

Figure 4.2: Histogram from Figure 4.1(b) smoothed

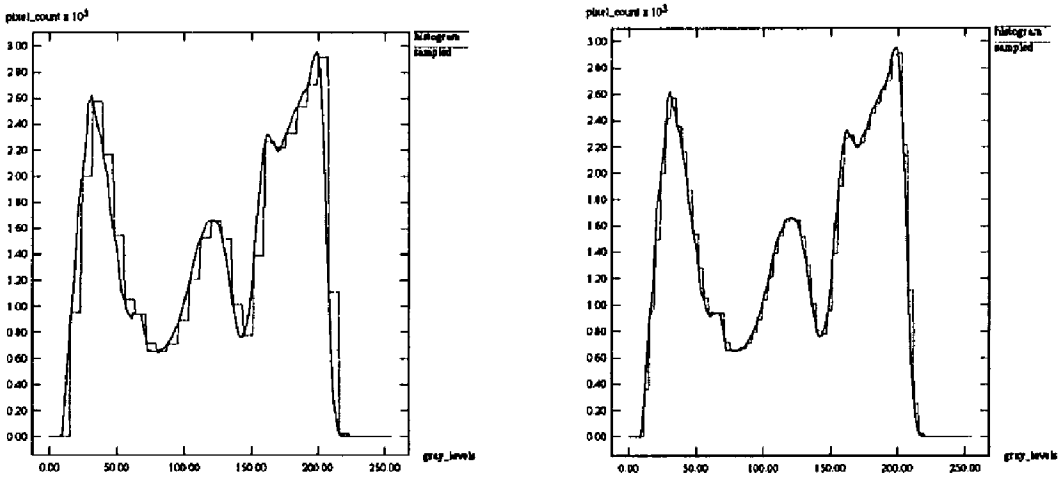
the original histogram by locating local minima in $\{a_k\}_j$ for increasing values of j corresponding to better and better approximations of f .

Before this method of successive approximations is applied, the original histogram is smoothed using the usual noise-removal techniques such as averaging and/or morphological filtering.

Figure 4.2 (a) shows the effect of morphological filtering with a window of size 3 on the histogram in Figure 4.1(b). Figure 4.2 (b) shows the same histogram after neighborhood averaging the result in (a) using a sliding window of about twice the size of the window used for morphological filtering. As mentioned before, three broad modes are now visible in the histogram. Figures 4.3 (a) through (d) show the effect of sampling this smoothed histogram at resolution levels -5, -4, -3, and -2 respectively. The number of modes in the sampled histogram increases with the level of resolution.



(a) j = -5 (b) j = -4



(c) j = -3 (d) j = -2

Figure 4.3: h_f^j sampled at different resolutions

(a) 2 regions (b) 3 regions

(c) 4 regions (d) 5 regions

Figure 4.4: Segmented images

Figure 4.4 shows the result of segmenting the original image (Figure 3.5 (a)) using the modality information gathered from the multiresolution decomposition of the histogram (Figure 4.3). No postprocessing was performed on these images to improve the quality of the segmentation. The vertical gray bands in Figures 4.4 (c) and (d) are not the result of bad segmentation; these artifacts (presumably the head signature from a device on which the original image was printed) are present in Figure 3.5 (a) and are easily discernible.

One other point is worth mentioning at this stage. The quality of the segmented image can be improved by utilizing pixel connectivity information in the image. Postprocessing using this proximity information has been deliberately avoided to show the results of the segmentation based solely on the multiresolution decomposition of the histogram. This holds for the images shown in section 4.6 also.

4.4 Algorithm $S1$

The coarsest resolution level at which h_j is sampled is dictated by modality considerations. Since at least two modes are required, at a particular resolution, for a threshold intensity level to be selected, the minimum number of sample points is 4. It is not possible to have a bimodal histogram with less than 4 sample points. Thus we have $j_{min} = -\log(\alpha/4)$ where α is the number of intensity levels in the image. Algorithm $S1$ analyzes h_j by analyzing the sequences $\{a_k\}_{j \geq j_{min}}$ for increasing values of j. An input to the algorithm is the minimum number of desired regions (R) in the segmented, labeled image. This number is used as the stopping critereon for the termination of the algorithm. The algorithm is given in Figure 4.5.

1. Construct the histogram h_f of the input image f
2. Low pass filter h_f to remove high-frequency noise
3. $j \leftarrow -\log_2(\alpha/4)$
4. $\{a_k\}_j = \{(h_f, \phi_{j,k}\}_{k \in Z}$
5. $n \leftarrow \mathcal{M}(\{a_k\}_j)$
6. If $n < R$, set $j \leftarrow j + 1$ and go to Step 4 if $j < 0$
7. Obtain a set of thresholds $\{\tau_l\}_{1 \leq l \leq n}$ from the modes of $\{a_k\}_j$
8. Label pixels in the image using $\{\tau_l\}$ i.e. label the pixel in the output image g at (x, y) with label j if $\tau_j \leq f(x, y) \leq \tau_{j+1}$; $\tau_0 = 0$ and $\tau_{n+1} = n + 1$.

Figure 4.5: Algorithm $S1$

In Step 2 of the above algorithm, low pass filtering is accomplished by morphological filtering followed by neighborhood averaging. For morphological filtering, a sliding one dimensional window of odd size is used to replace each element of the histogram by the maximum value in the window. Similarly, neighborhood averaging is performed by sliding a similar window over the data and replacing the center of the window by the average of the data values occurring in the window. The overall effect is that of suppressing the high frequency effects and retaining the low frequency components in the data.

In Step 5, $\mathcal{M}$ is function that analyzes the modes in the sequence $\{a_k\}_j$. For example, it could be a simple technique such as $\mathcal{M}(\{a_k\}_j) = \#\{l \mid a_{l-1} > a_l \wedge a_{l+1} > a_l\}$ where $\#$ denotes the cardinality operator on sets.

The algorithm makes at most two passes over the image data; the first pass is required for collecting the histogram data, and a final pass for labeling image pixels

with the thresholds detected in the sampled histogram. As a result, the algorithm is faster than segmentation methods that make multiple passes over the image. Note that the size of the histogram depends on the number of levels of quantization of the image and not on the size of the image. Thus, the computational complexity of processing the histogram is not included in the estimation of the overall complexity of the algorithm.

4.5 Algorithm $S2$

Algorithm $S1$ treats an image histogram as a static object in the sense that the smoothing (morphological filtering and averaging) operation is performed on the original histogram only once and thereafter, this smoothed version is analyzed in a multiresolution fashion for peaks and valleys. In this section, we describe an improved algorithm for segmenting an image by analyzing the modes in the projections of the histogram onto successively finer subspaces of $L^2(\mathbf{R})$ using wavelets.

Let $\psi_{j,k}; j,k \in Z$ be a wavelet orthonormal basis of $L^2(\mathbf{R})$. There exist subspaces of $L^2(\mathbf{R})$, W_j which are spanned by $\{\psi_{j,k}\}_{k\in Z}$. Recall that $\psi_{j,k} = 2^{j/2}\psi(2^j x - k)$ are the scaled and translated copies of the *mother wavelet* ψ. Now, rewriting equation 2.12 as

$$L^2(R) \;=\; V_0 \oplus \bigoplus_{j=0}^{\infty} W_j \;, \tag{4.8}$$

it is possible to analyze any function in $L^2(\mathbf{R}^2)$ by studying its projections onto V_0 and $W_j; j \geq 0$. That is, if P_0 and Q_j are the projection operators onto V_0 and W_j respectively, any function h_f in $L^2(\mathbf{R})$ may be written as

$$h_f \;=\; P_0(h_f) + \sum_{j=0}^{\infty} Q_j(h_f) \;. \tag{4.9}$$

Thus,

$$h_f = \sum_k a_k \, \phi_{0,k} + \sum_j \sum_k b_{j,k} \, \psi_{j,k} \tag{4.10}$$

where $\{a_k\} = \{\langle h_f, \phi_{0,k} \rangle\}$ and $\{b_{j,k}\} = \{\langle h_f, \psi_{j,k} \rangle\}$ and the summation involves only finitely many terms due to the finiteness of the sampling of h_f as discussed earlier. In the above equation, setting some (or all) of the coefficients $b_{j,k}$ to zero yields an approximation $\tilde{h}_f$ to h_f (in the L^2 sense) which has the following property

$$\|\tilde{h}_f - h_f\|^2 \leq \|g_f - h_f\|^2 \tag{4.11}$$

for any g_f in $L^2(\mathbf{R})$ which has the same set of zero wavelet coefficients as $\tilde{h}_f$. Therefore, setting the wavelet coefficients $b_{j,k}$ to zero yields the most faithful approximation to the original function in the L^2 sense.

The main advantage of this approach over the method followed in algorithm $S1$ is the fact that, at each segmentation stage, the smoothed version of the original histogram is recomputed as a projection onto a finer subspace of $L^2(\mathbf{R})$. In $S1$, the histogram was smoothed only once; here it is smoothed from the original (finest scale) histogram by discarding fewer and fewer wavelet coefficients. The algorithm follows. Note that, if the histogram has α gray levels, its wavelet transform will have one scaling function coefficient, and $\log_2(\alpha)$ levels of wavelet coefficients. As in algorithm $S1$, R is the minimum number of desired regions in the final segmented image, and is an input parameter to algorithm $S2$. The algorithm $S2$ is shown in Figure 4.6.

1. Construct the histogram h_f of the input image f
2. $\hat{h}_f \leftarrow \mathcal{W}h_f$ i.e. $\hat{h}_f = \{\{a_k\}, \{b_{j,k}\}\}$ are the wavelet coefficients of h_f
3. $j \leftarrow 0$
4. $\{d_{j,k}\} \leftarrow \{b_{j,k}\}$
5. $\forall l, l \geq j,\ d_{j,k} \leftarrow 0$
6. $\tilde{h}_f \leftarrow \sum_k a_k\ \phi_{0,k} + \sum_j \sum_k d_{j,k}\psi_{j,k}$
7. $n \leftarrow \mathcal{M}(\tilde{h}_f)$
8. If $n < R$, set $j \leftarrow j + 1$ and go to Step 4 if $j < \log_2(\alpha)$
9. Obtain a set of thresholds $\{\tau_l\}_{1 \leq l \leq n}$ from the modes of $\tilde{h}_f$
10. Label pixels in the image using $\{\tau_l\}$ i.e. label the pixel in the output image g at (x, y) with label j if $\tau_j \leq f(x, y) \leq \tau_{j+1}$; $\tau_0 = 0$ and $\tau_{n+1} = n + 1$.

Figure 4.6: Algorithm $S2$

In Step 2 of the algorithm, $\mathcal{W}h$ is the discrete one-dimensional wavelet transform of the histogram function. Steps 5 and 6 have the combined effect of low pass filtering the histogram. As j increases, the size of the passband in the frequency domain also increases i.e. features corresponding to higher and higher frequencies in the frequency domain are retained in the reconstructed histogram h_f.

As in algorithm $S1$, only two passes over the input image are required, once for constructing the image histogram and finally for the labeling of the segmented image using the thresholds determined by the algorithm. Again, the cost of processing the image histogram is independent of the size of the input image. As a result, the complexity of the algorithm is linear in the size of the input image. The next section presents results of applying algorithms $S1$ and $S2$ to some test images.

Table 4.1: Summary of results obtained with algorithm $S1$

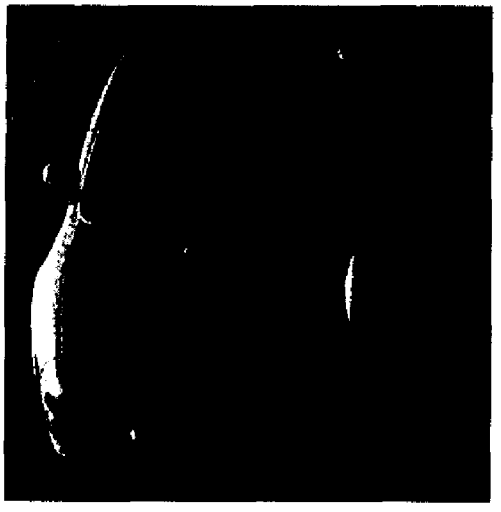
	# of regions	
image	$j = -5$	$j = -4$
1 (Fig. 4.4)	2	3
2 (Fig. 4.7)	3	5
3 (Fig. 4.8)	4	6

4.6 Results

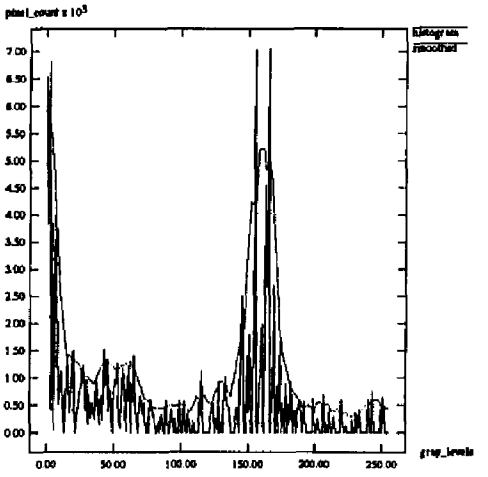
We have applied the algorithms described in the previous sections to several test images. All test images had multimodal histograms. As mentioned earlier, no post processing operations were performed on the segmented images to improve the quality of segmentation. No attempt was made to select test images that were free of texture. However, images with a significant amount of texture information would not be segmented satisfactorily using the proposed algorithms since textured images don't display well-defined modes in their histograms at most scales. In the remainder of this section, we present a small sample of test images and the results of applying algorithms $S1$ and $S2$.

Figures 4.4, 4.7 and 4.8 show the results of applying algorithm $S1$. Table 4.1 summarizes the results obtained with this algorithm.

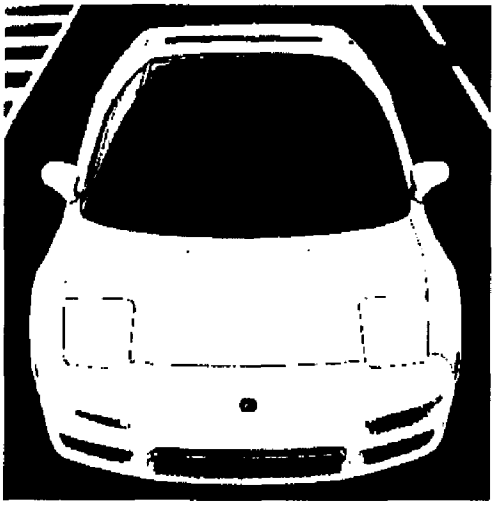
Segmentation of satellite images is a problem of significant importance in oceanography. For instance, the detection of mesoscale features [33] is useful for understanding ocean dynamics. Figures 4.9, 4.10, 4.11, and 4.12 show the performance of the proposed segmentation algorithms on satellite images. The feature of interest is the gulf stream which is a current of relatively warm water amidst other cold water bodies in the Atlantic Ocean.



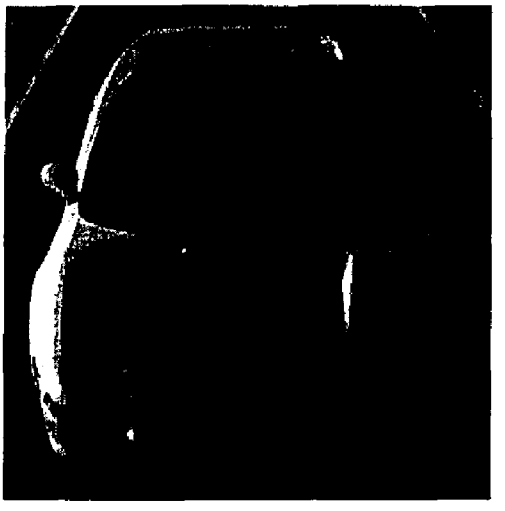
(a) original image



(b) histogram

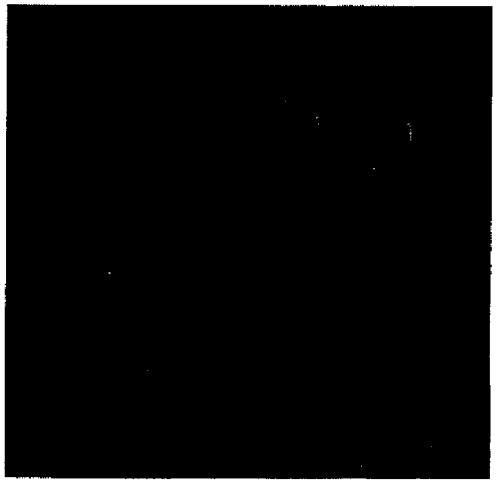


(c) 3 regions at j = -5

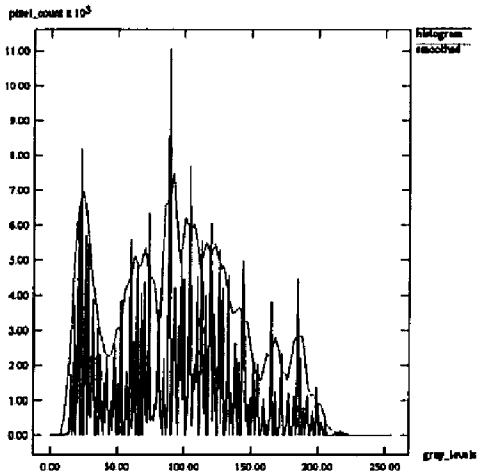


(d) 5 regions at j = -4

Figure 4.7: Results of applying the algorithm S1



(a) original image



(b) histogram

(c) 4 regions at j = -5

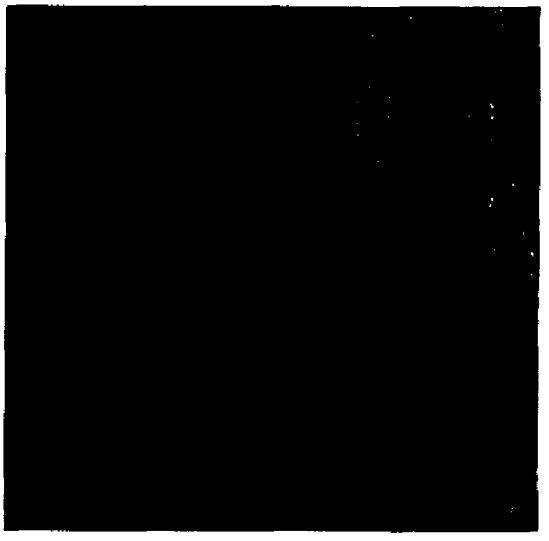
(d) 6 regions at j = -4

Figure 4.8: Results of applying the algorithm S1

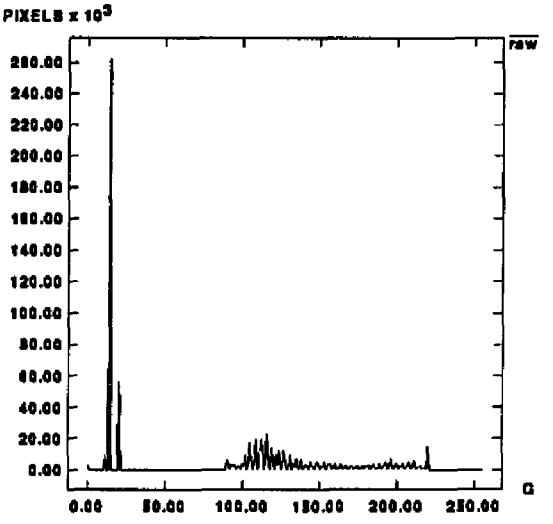
In the original image ((a) in the four figures), the gulf stream shows up as a dark gray band oriented along the southwest–northeast diagonal. (b) in each figure shows the histogram of intensity levels detected in the image, and this statistic is not particularly enlightening; all we may gather from the histogram is that there is a sharp peak near the intensity level 15 which probably corresponds to the large dark land mass to the left of the image. Looking closely at the envelope of the histogram, we do find three broad modes.

The gulf stream, being dark gray in color, is most likely to be buried in the mode in the middle of the histogram. Algorithms $S1$ and $S2$ zoom in to the gray levels representative of the gulf stream and isolate it. In Figure 4.11 (c), algorithm $S1$ failed to locate the gulf stream while algorithm $S2$ succeeded. This slight edge $S2$ has over $S1$ is due to the fact that the wavelet based smoothing approach is mathematically more sound than the averaging technique.

Table 4.2 summarizes the results obtained from the two algorithms proposed. The results for algorithm $S1$ correspond to a projection of the smoothed histogram onto a coarse subspace at resolution $j = -3$ for all images. Algorithm $S2$ used the Daubechies 18-tap wavelet filter for smoothing the histogram. The column labeled "# regions" lists the number of segmented regions in the final image. This number is greather than the number of thresholds located in the processed histogram by unity. The column labeled "Gulf stream?" indicates whether the algorithm was successful in isolating the Atlantic Gulf Stream as a separate region. Note that the performance of $S2$ is consistently better than that of $S1$. $S2$ is able to better isolate the feature of interest (the Gulf Stream) with a smaller number of regions than $S1$. In the case of image 3 (Fig . 4.11), with a final segmented image that



(a) original image (256 levels)

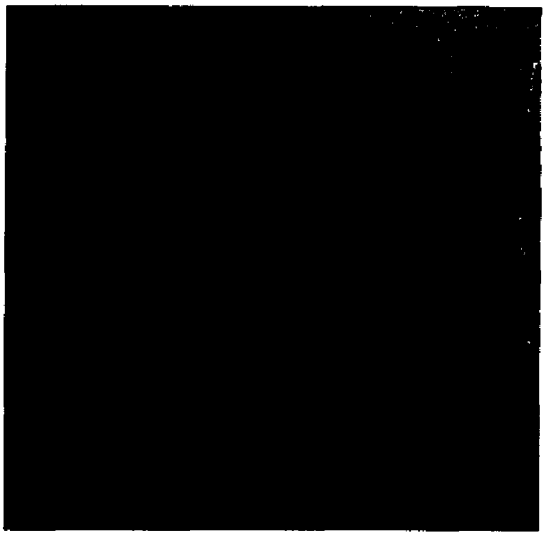


(b) raw image histogram

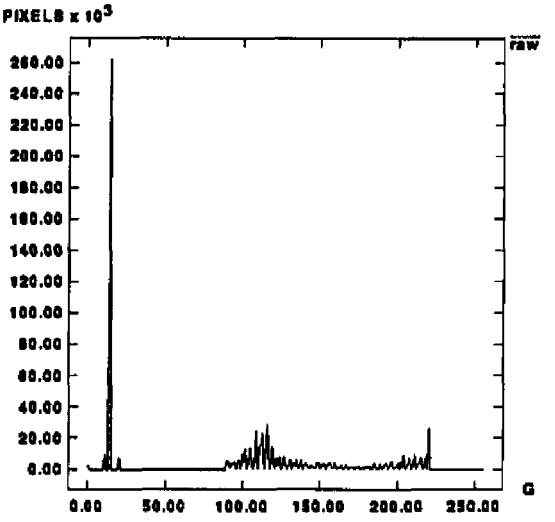
(c) algorithm $S1$ (6 regions)

(d) algorithm $S2$ (4 regions)

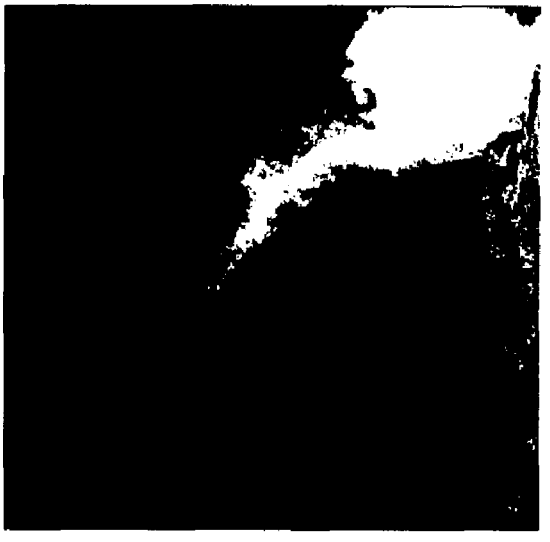
Figure 4.9: Comparison of segmentation schemes: Image 1



(a) original image (256 levels)



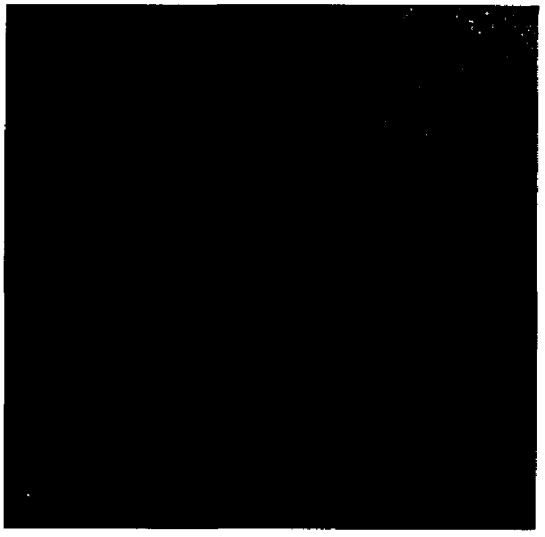
(b) raw image histogram



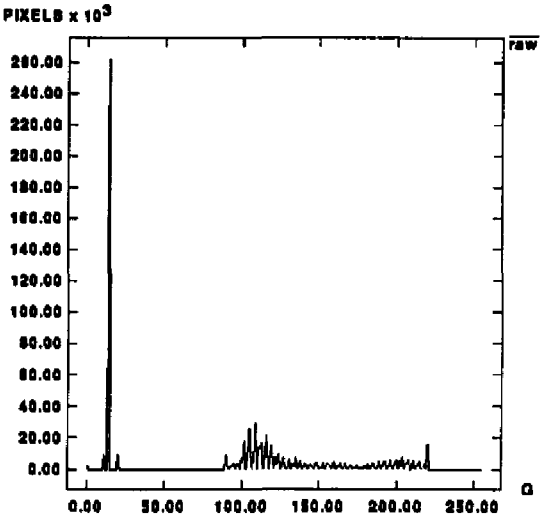
(c) algorithm $S1$ (6 regions)

(d) algorithm $S2$ (4 regions)

Figure 4.10: Comparison of segmentation schemes: Image 2



(a) original image (256 levels)

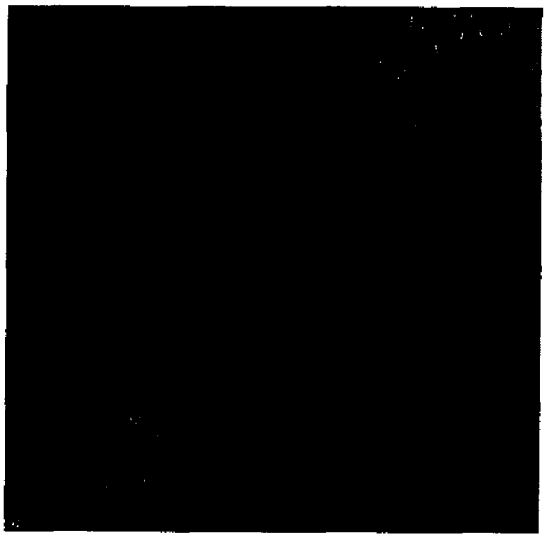


(b) raw image histogram

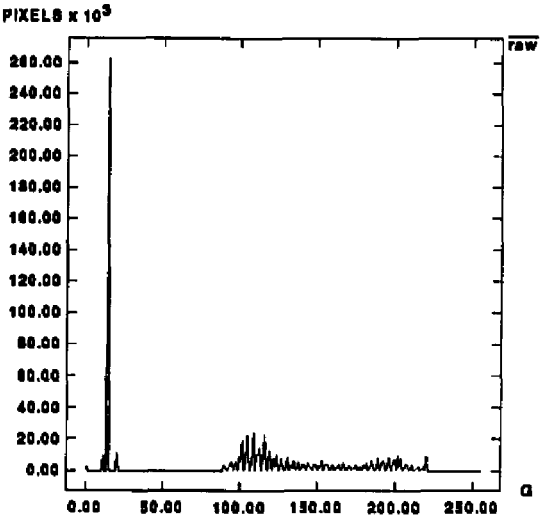
(c) algorithm $S1$ (4 regions)

(d) algorithm $S2$ (4 regions)

Figure 4.11: Comparison of segmentation schemes: Image 3



(a) original image (256 levels)



(b) raw image histogram

(c) algorithm $S1$ (6 regions)

(d) algorithm $S2$ (4 regions)

Figure 4.12: Comparison of segmentation schemes: Image 4

Table 4.2: Summary of segmentation results

image	Algorithm $S1$		Algorithm $S2$	
	# regions	Gulf stream ?	# regions	Gulf stream ?
1 (Fig. 4.9)	6	yes	4	yes
2 (Fig. 4.10)	6	yes	4	yes
3 (Fig. 4.11)	4	no	4	yes
4 (Fig. 4.12)	6	yes	4	yes

contains 4 regions, $S1$ is unable to locate the Gulf stream while $S2$ locates the feature successfully.

In conclusion, it is relevant to observe that the shape of the projected histogram in the case of algorithm $S2$ is sensitive to the analyzing wavelet used. For example, the smoother the wavelet used, the smoother the projected histogram. If a very short filter (for instance, the Daubechies 4-tap filter) is used, the local extrema in the projected histogram do not appear smooth due to the jagged nature of the wavelet functions themselves. As a result of this, selection of thresholds becomes a problem since the valleys are not well defined in the histogram. Experiments with the Daubechies wavelets of various filter lengths show that, in general, the longer the wavelet, the better the coarse segmentation.

It is also relevant to mention here that images with large amounts of texture and images whose histograms do not show any modality properties at all (even in their envelopes) will not be segmented satisfactorily using algorithms $S1$ and $S2$ since these algorithms depend on the modes of the histograms, at some fine or coarse level, for the selection of thresholds. Thus, these algorithms will not be useful for segmentation of texture fields in an image, for example.

Chapter 5

Edge Detection

Edge detection [63] is a fundamental problem in image processing. The presence of an edge in an image signals the location of an object or a feature of interest. Edge detection is, in some sense, the conceptual dual of the problem of segmentation. The overall objective, in the problem of edge detection, is to isolate (and label) pixels in an image where a sudden change in the intensity level occurs. If the distribution of gray levels in an image is assumed to follow some smooth surface, an edge is represented as a discontinuity in this surface. Our main concern in this chapter (and this dissertation) will be to develop edge detection schemes which operate successfully on satellite images which are characterized by weak gradients.

Conventional edge detectors such as the Sobel operator, the Roberts operator and other similar operators (see Section 5.3.1) are of limited use when applied to satellite images (such as those shown in Figures 4.8–4.10 (a)) since they are too sensitive to noise and respond very weakly to the edges of interest such as those around the Atlantic Gulf Stream. Thus, an edge enhancement scheme is required for isolating edge pixels in the image.

The following two sections describe two new approaches to the problem of edge pixel detection. Algorithm $E1$ uses the projection of the original image onto appropriate subspaces of $L^2(\mathbf{R})$ to extract edge pixel candidates and their relative magnitudes in a reliable fashion.

Algorithm $E2$ uses the segmentation algorithms developed in the previous chapter to identify pixels that lie on the borders between adjacent regions. Both algorithms are thus edge detectors in the sense that they identify pixels in the image that are candidates for being edge pixels. Thus, these two algorithms are, in some sense, preprocessing algorithms which identify pixels in the image which have a strong chance being edge pixels. For an estimate of the edge strength at these points, however, a conventional derivative based method may be employed.

5.1 Algorithm $E1$

An edge is an intensity level discontinuity in an image. Thus, an edge is said to be present whenever a sudden change in intensity levels occurs over a small spatial scale in the image. Therefore, at a sufficiently fine scale, the convolution of the wavelet basis function (at that scale) with the image shows an extremum very close to the actual edge. At coarser scales, since the support of the wavelet basis function itself spans (possibly) several pixels at the fine scale, the resoponse of the basis function looks more like band and less like a line. Intuitively therefore, a good approximation to the edge map of an image may be obtained by convolving the basis functions at a sufficiently fine scale with the (low pass filtered) image. In fact, Mallat and Zhong [46] have developed an algorithm for coding an image economically using multi scale edges obtained by separating the input image in an orthogonal wavelet basis.

The basic idea behind algorithm $E1$ is the high pass filtering of the input image using orthogonal wavelet filters followed by a logarithmic transformation of the high pass data which ensures that weak edges in the input image are enhanced.

Algorithm $E1$

1. Expand the image I into a two-dimensional basis i.e.

$$I(x,y) = \sum_k \sum_l s_{k,l}\phi_k\phi_l + \sum_{j=0}^{n-1} \sum_k \sum_l d^{\phi\psi}_{j,k,l}\phi^j_k\psi^j_l + d^{\psi\phi}_{j,k,l}\psi^j_k\phi^j_l + d^{\psi\psi}_{j,k,l}\psi^j_k\psi^j_l$$

2. $\forall j,k,l s_{k,l} \leftarrow 0$

3. $\forall j, j < L, d_{j,k,l} \leftarrow 0$

4. Reconstruct an approximation $\tilde{I}$ to I using the remaining wavelet coefficients i.e. $(j < L)$

$$\tilde{I}(x,y) = \sum_k \sum_l s_{k,l}\phi_k\phi_l + \left\{ \sum_k \sum_l d^{\phi\psi}_{j,k,l}\phi^j_k\psi^j_l + d^{\psi\phi}_{j,k,l}\psi^j_k\phi^j_l + d^{\psi\psi}_{j,k,l}\psi^j_k\psi^j_l \right\}_{j=L}$$

5. Construct the edge map image $\tilde{I}_E$ by applying a logarithmic transformation to the reconstructed image $\tilde{I}$; i.e. $\tilde{I}_E(x,y) = \log(1 + \tilde{I}(x,y)$, and quantize $\tilde{I}_E$ to the dynamic range of intensity levels in the input image for better resolution.

End Algorithm $E1$

Figure 5.1: Algorithm $E1$

For the best localization of the edge in the spatial domain, a small filter is required. Using the Haar wavelet for instance, localizes an ideal edge to one pixel; that is, the response of the wavelet peaks at exactly one pixel. The algorithm is given below. L in Step 3 is the number of levels in the wavelet decomposition of the image. In Step 4, the high pass component of the input image is computed by reconstructing the image from the wavelet coefficients at the finest scale and discarding all other coefficients.

Table 5.1: Maximum edge strength

image	maximum	log(maximum)
Fig. 5.2(a)	96.25	4.57
Fig. 5.2(b)	146.75	4.99
Fig. 5.2(c)	85.25	4.45

Table 5.2: Maximum edge strength

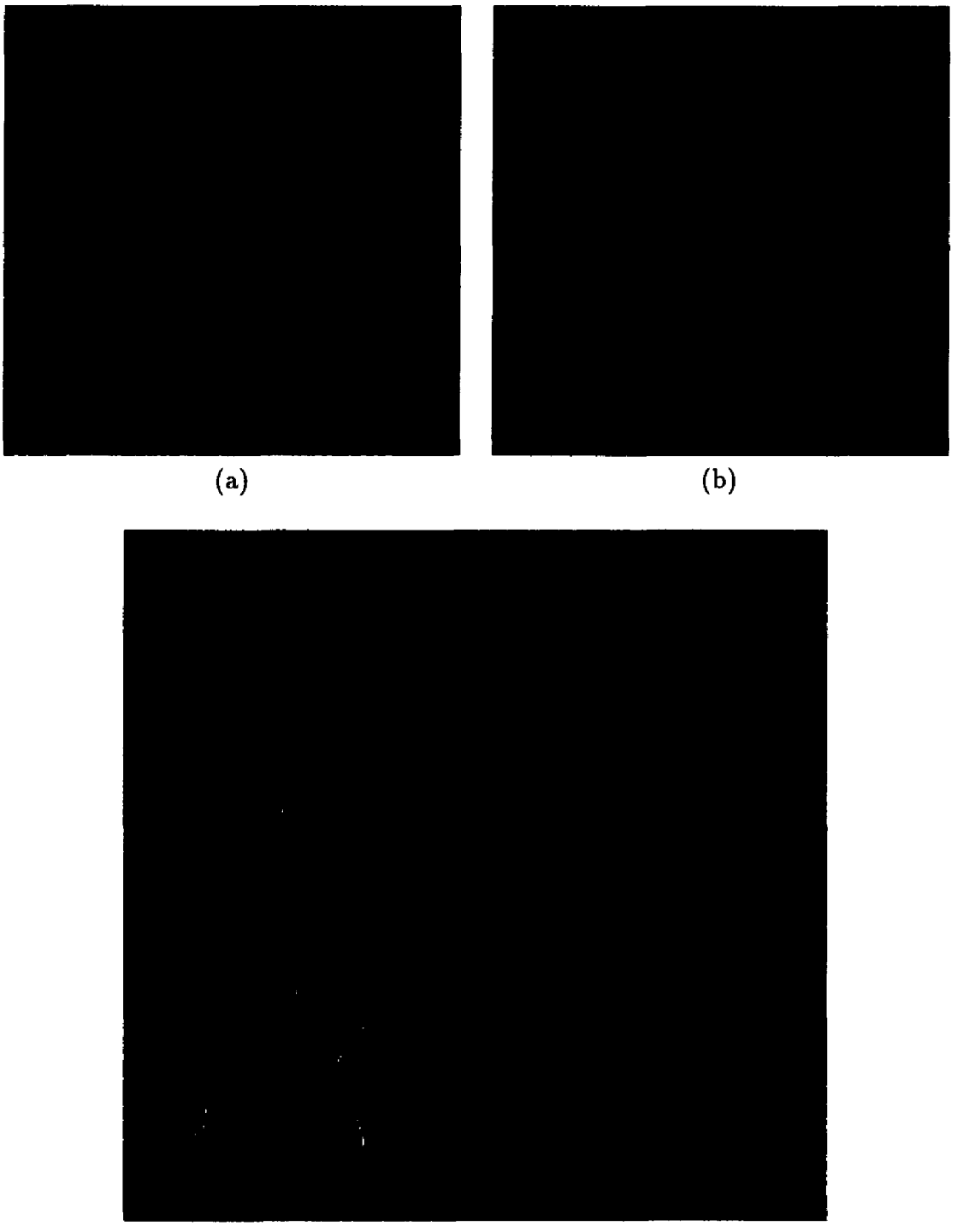
image	maximum	log(maximum)
Fig. 5.3(a)	157.25	5.06
Fig. 5.3(b)	158.00	5.06
Fig. 5.3(c)	161.00	5.08
Fig. 5.3(d)	161.00	5.08

Figure 5.2 shows the results of applying algorithm $E1$ to sample images. These images do not have weak gradients which are typically found in satellite images (see figure 5.3). The wavelet used to analyze the images was the Haar wavelet which was chosen on account of its small filter length.

Table 5.1 gives the maximum edge strength values and the log (to base e) of the maximum edge strength values found in the three test images shown in Figure 5.2.

Figure 5.3 shows the results of applying algorithm $E1$ to some satellite images which typically have weak gradients. The feature of interest is the Atlantic Gulf stream which appears as a very weak edge (especially its south wall) parallel to the south-west — north-east direction.

Table 5.2 summarizes the maximum edge strength values found in the test images shown in Figure 5.3. To better isolate the Gulf stream, an edge length thresholding scheme may be used.



(a) (b)

(c)

Figure 5.2: Results of applying $E1$ to images without weak gradients

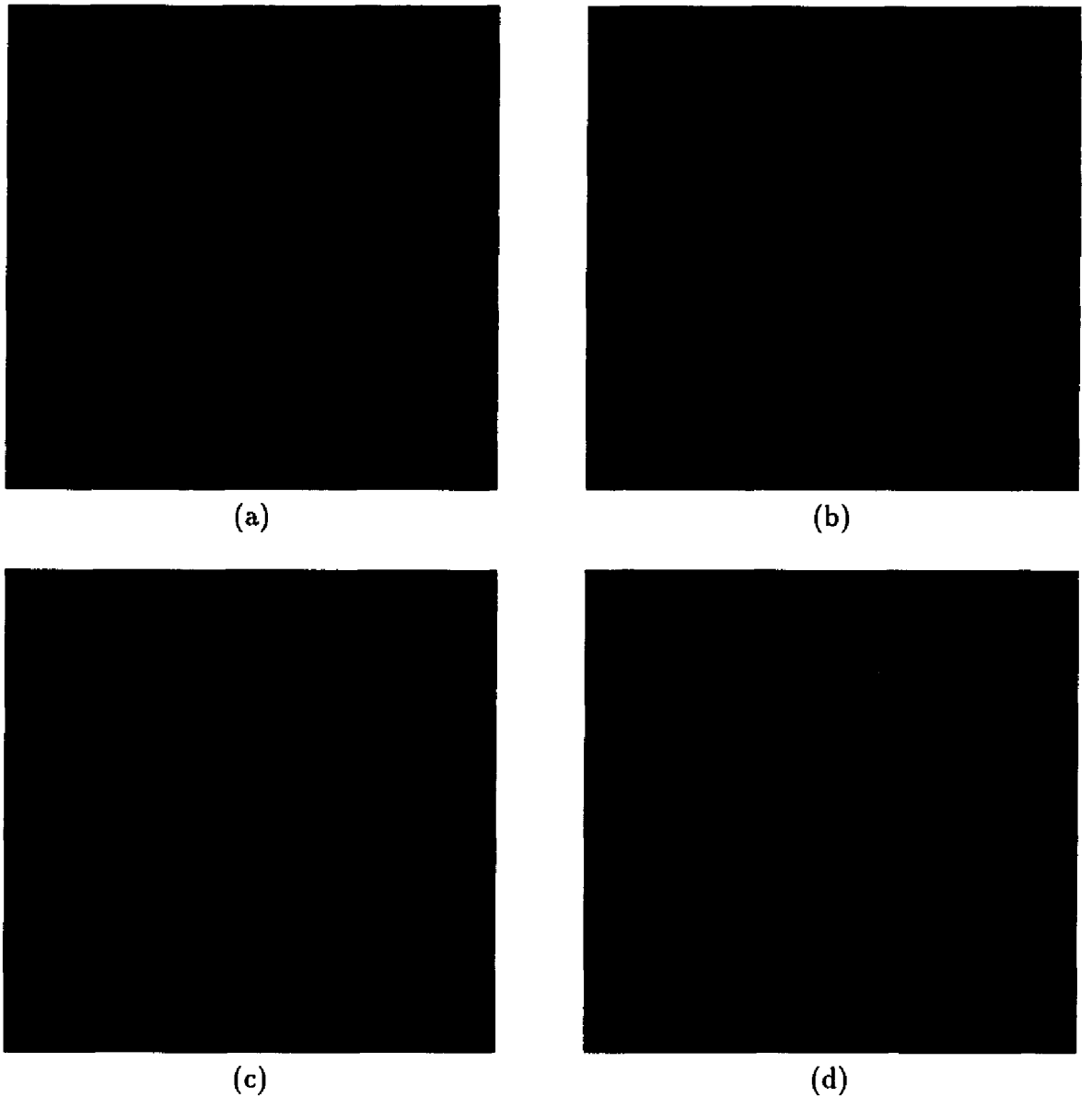


Figure 5.3: Results of applying $E1$ to images with weak gradients

5.2 Algorithm $E2$

The basic idea in this algorithm involves the segmentation of the original image into coarse regions and identifying the boundaries between these regions as edges. As the segmentation of the original image is refined, the edge image is also correspondingly refined. The advantage of the initial segmentation process is the improved separation (in gray scale space) between regions with very similar gray levels. Two different versions of the basic idea may be implemented as described below.

5.2.1 Version 1

As mentioned in the beginning of this chapter, we are mainly concerned with edge detection in images with weak gradients. For instance, the edges in an infrared image between adjacent bodies of water in the Atlantic ocean, due to differences in water temperature, are not as strong as the land-sea edge. As a result, any algorithm that treats all edges in the image uniformly, has limited success in isolating the fine edges. If a derivative operator is used to evaluate the edge strength, a very low threshold value is required before the water-water edge is recognized. But, at such low threshold values, noise pixels in the image are also recognized as edge candidates, and some sort of a post-processing algorithm has to be applied to the edge image to remove noise.

Let I be an image which is to be analyzed for edges. Let I_l and I_h be the low-pass images and high-pass images respectively, obtained from a reconstruction of the two-dimensional wavelet transform of I. Now, I_h is a projection of the image

onto the union of the detail spaces at the finest resolution. Thus, local extrema in I_h correspond to edges in the image (for details see [46]).

High-frequency noise in an image manifests itself as spurious edges in the detail images. Filtering out this noise, unfortunately, leads to loss of edge detail as well since a fraction of the edges in the detail images do correspond to *bona fide* edges. Clearly, any form of filtering which discards information, from any frequency band, is unable to distinguish between noise and signal. Observing that satellite images display edges of interest that survive across several scales, this algorithm performs low-pass filtering using the wavelet transform of the image before extracting edges. The algorithm follows (L in Step 2 is a parameter to the algorithm which determines the amount of low-pass filtering).

Step 6 in the algorithm above is very sensitive to the parameters used in the segmentation (as described in algorithm $S2$). Fortunately, satellite images are tolerant to a wide range of parameters since they display well defined regions. Figures 5.5, 5.6, 5.7 and 5.8 show the results of applying algorithm $E1$ to a set of sample satellite images. In each figure, (a) shows the original image and the other three images show the edges detected for various values of L in Step 2 above. The feature of interest, the Atlantic Gulf Stream is clearly visible in the edge images.

5.2.2 Version 2

This version is again implemented as a preprocessing operation, and thus, any edge operator may be applied to the resulting image. The algorithm is simply described as follows.

Algorithm $E2$ (version 1)

1. Expand the image I into a two-dimensional basis i.e.

$$I(x,y) = \sum_k \sum_l s_{k,l}\phi_k\phi_l + \sum_j^{n-1} \sum_k \sum_l d^{\phi\psi}_{j,k,l}\phi^j_k\psi^j_l + d^{\psi\phi}_{j,k,l}\psi^j_k\phi^j_l + d^{\psi\psi}_{j,k,l}\psi^j_k\psi^j_l$$

2. $\forall j, j > L, d_{j,k,l} \leftarrow 0$

3. Reconstruct an approximation $\tilde{I}$ to I using the remaining wavelet coefficients i.e. $(j < L)$

$$\tilde{I}(x,y) = \sum_k \sum_l s_{k,l}\phi_k\phi_l + \sum_j^{L} \sum_k \sum_l d^{\phi\psi}_{j,k,l}\phi^j_k\psi^j_l + d^{\psi\phi}_{j,k,l}\psi^j_k\phi^j_l + d^{\psi\psi}_{j,k,l}\psi^j_k\psi^j_l$$

4. Compute the histogram $h_{\tilde{I}}$ of $\tilde{I}$

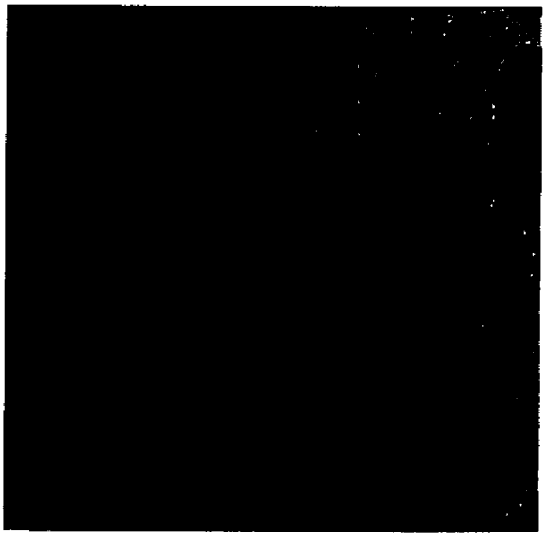
5. Project the histogram of $\tilde{I}$ onto a set of subspaces of $L^2(\mathbf{R})$ as described in algorithm $S2$

6. Segment the lowpass image $\tilde{I}$ using the multiresolutional modes obtained from the projections of $h_{\tilde{I}}$

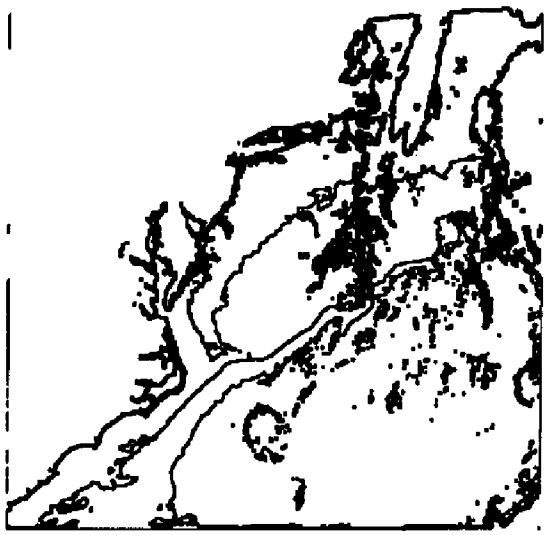
7. Label pixels in the segmented image that lie on the borders between adjacent segmented regions as candidates for edge pixels

End Algorithm $E2$ (version 1)

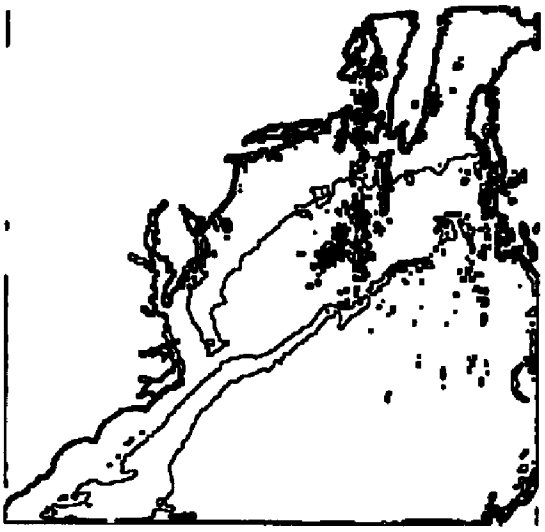
Figure 5.4: Algorithm $E2$ (version 1)



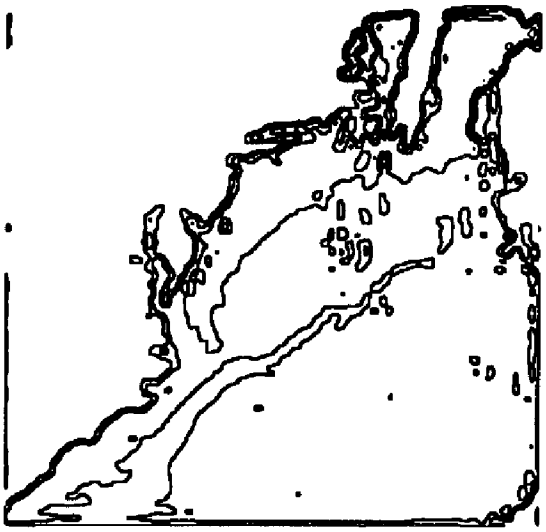
(a) original image



(b) edge image ($L = 1$)

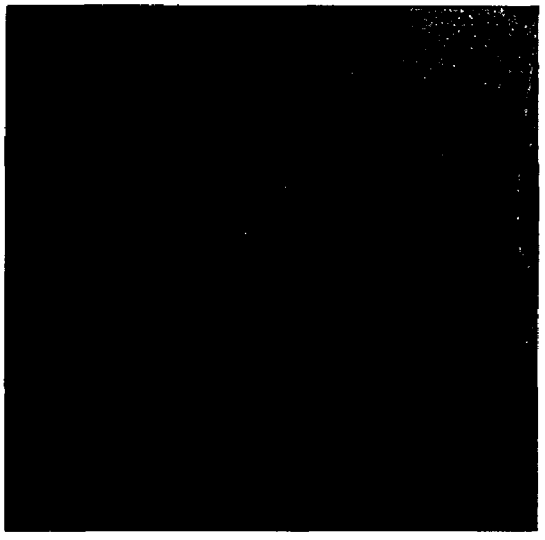


(c) edge image ($L = 2$)

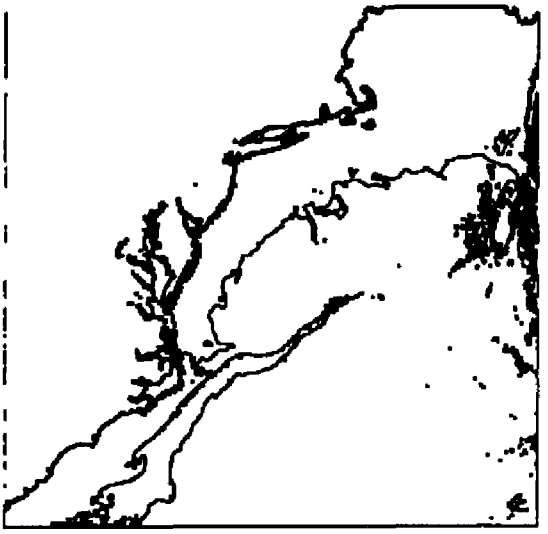


(d) edge image ($L = 3$)

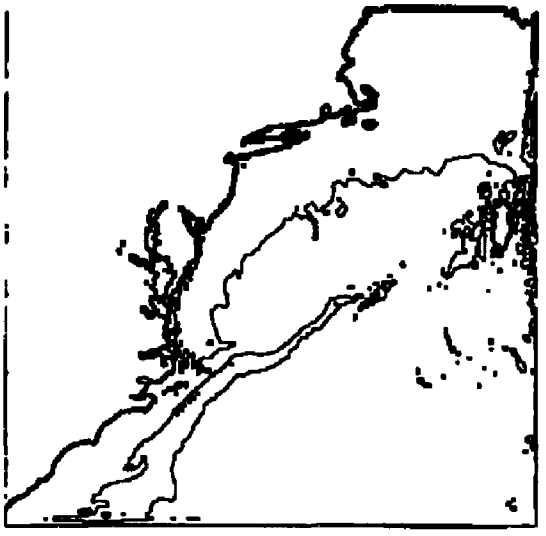
Figure 5.5: Result of applying algorithm $E2$ version 1



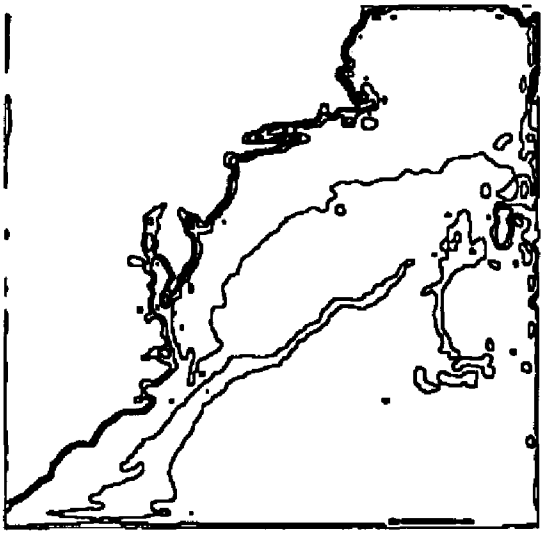
(a) original image



(b) edge image ($L = 1$)

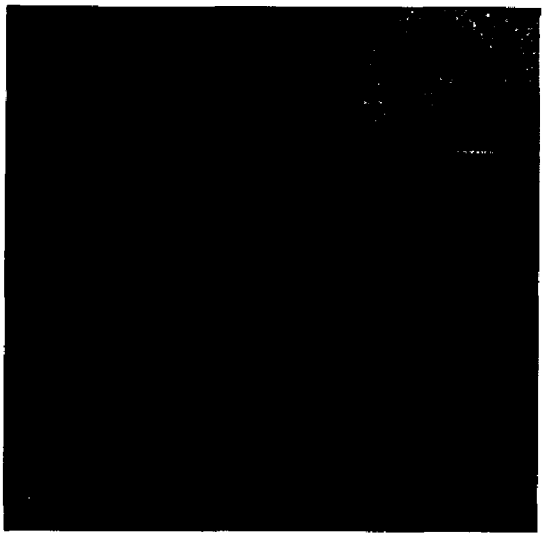


(c) edge image ($L = 2$)

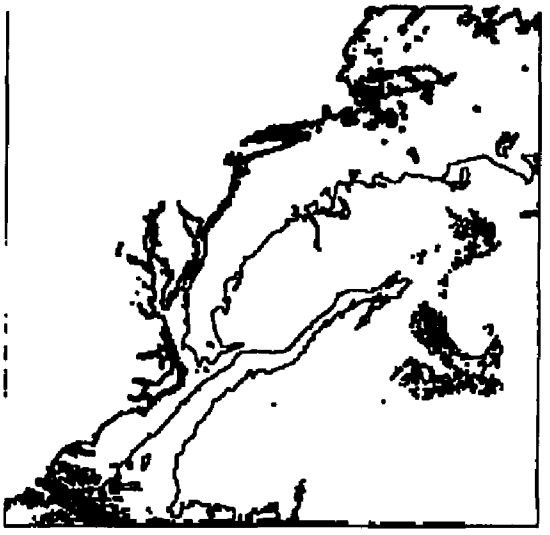


(d) edge image ($L = 3$)

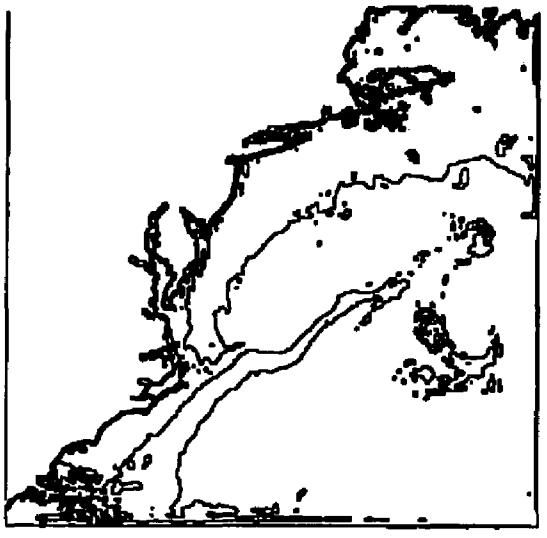
Figure 5.6: Result of applying algorithm $E2$ version 1



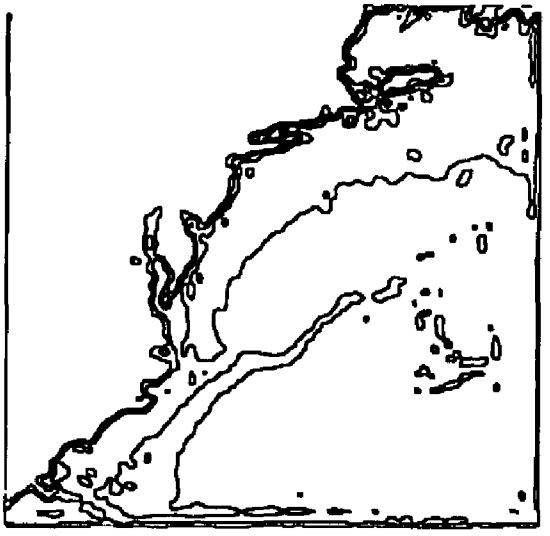
(a) original image



(b) edge image ($L = 1$)

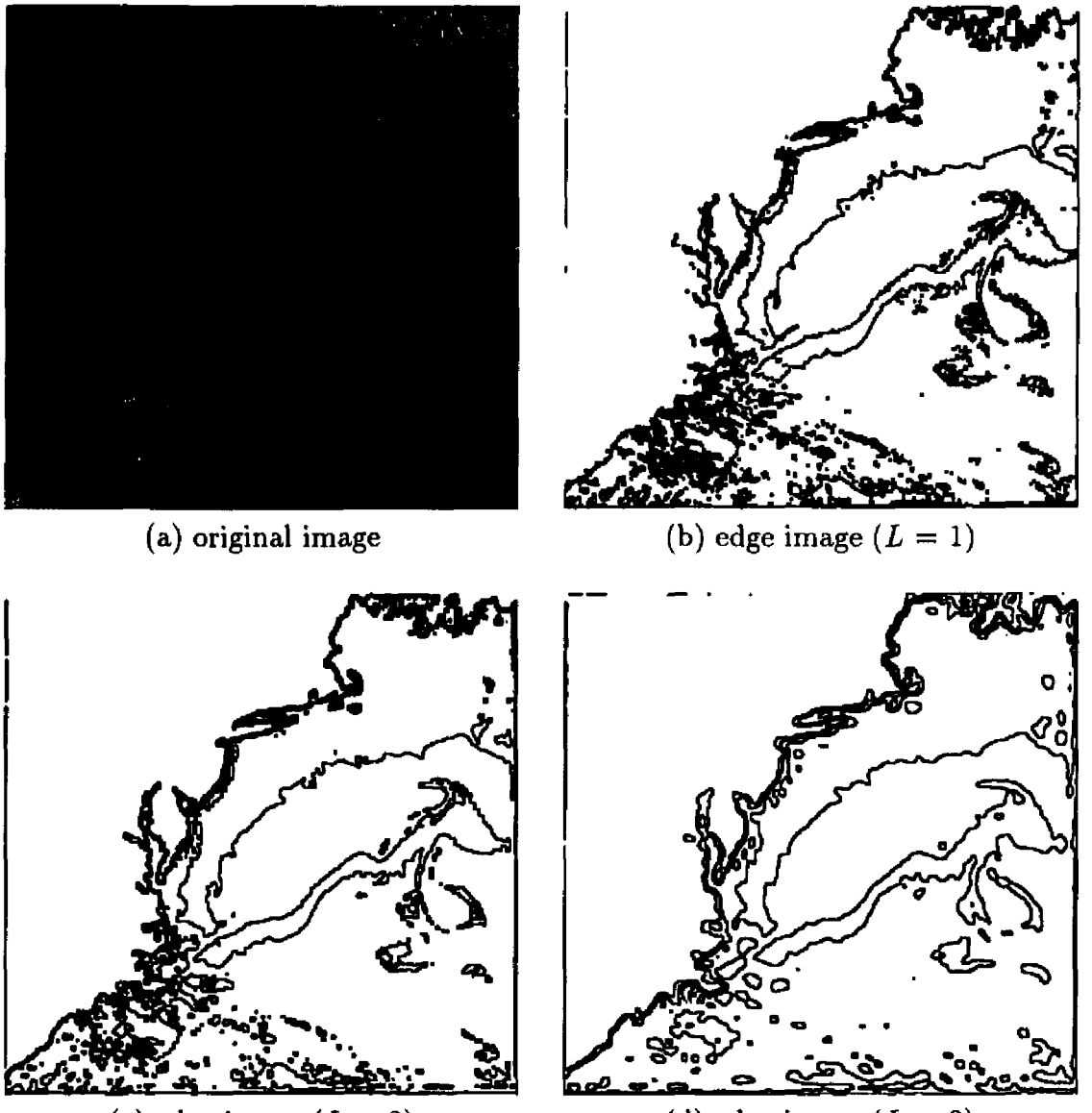


(c) edge image ($L = 2$)



(d) edge image ($L = 3$)

Figure 5.7: Result of applying algorithm $E2$ version 1



(a) original image

(b) edge image ($L = 1$)

(c) edge image ($L = 2$)

(d) edge image ($L = 3$)

Figure 5.8: Result of applying algorithm $E2$ version 1

Algorithm $E2$ (version 2)

1. Smooth the input image I to remove high-frequency noise
2. Compute the histogram h_I of I
3. Apply a multiresolution decomposition algorithm ($S1$ or $S2$) to h_I for a segmentation of the image
4. Apply a morphological opening operation (see section 5.3.2) to the segmented image to clean the segmented image
5. Label pixels lying on the boundaries between adjacent regions as edge pixel candidates.

End Algorithm $E2$ (version 2)

Figure 5.9: Algorithm $E2$ (version 2)

In the above algorithm, Step 3 is the critical component. Depending upon the quality of the segmentation obtained at this stage, the edge labeling in Step 5, which is deliberately kept simple, performs satisfactorily or poorly. A good heuristic to use, in the absence of any *a priori* information about the class of images being dealt with, is to perform a conservative segmentation — one that leads to a large rather than a small number of regions. Subsequently, the quality of edges obtained can be adjusted finely by controlling the size of the morphological structuring element in Step 4. Alternatively, an edge-length based thresholding step may be appended to the above algorithm to filter out small edges while retaining long edges. Figures 5.10 and 5.11 show the results of applying algorithm $E2$ to a few test images. These test images have been chosen to show the performance of the algorithm in the presence

of weak gradients. For a comparison of the performance of this algorithm with conventional edge operators, see the next section.

The computational complexity of Step 1 in the algorithm (as already described in Chapter 4) is linear in the image size. At most two passes are required over the image. Steps 2 and 3 require at most one pass each over the input image. Thus the overall complexity is $O(n^2)$ where n is the linear dimension of the input image.

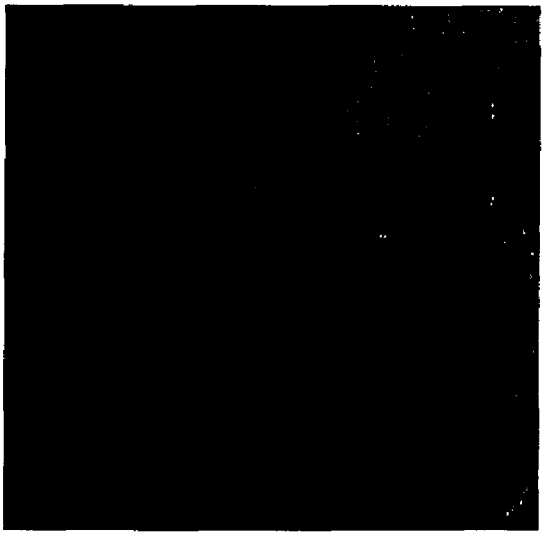
5.3 Conventional Detectors

5.3.1 Derivative Operators

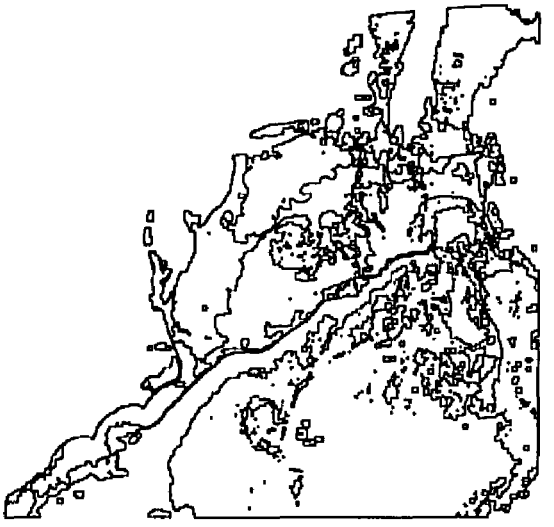
An edge is a boundary between two regions which are uniform with respect to intensity levels. Thus, a good measure of 'edginess' is the magnitude of the response of a local derivative operator at each pixel in the image. If the edge is modeled as a smooth (although rapidly rising/falling) transition from one homogeneous region to another with different gray level properties, the first derivative operator displays an extremum value at center of this transition. Corresponding to this extremum value of the first derivative, the second derivative displays a zero-crossing, also centered at the center of the edge transition. The Marr-Hildreth edge operator [47] relies on the second derivative response for isolating edges while the Canny operator [8] uses the first derivative.

The Sobel edge operator consists of two convolution kernels as shown in Figure 5.12. The kernel shown in (a) is sensitive to horizontal edges while (b) is sensitive to vertical edges.

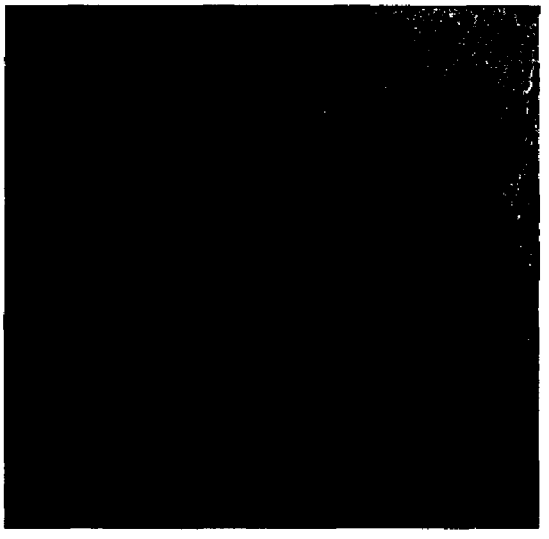
The output of the Sobel edge operators to a typical satellite image is shown in Figure 5.13. (a) and (b) show the output of S_h and S_v respectively. (c) and (d)



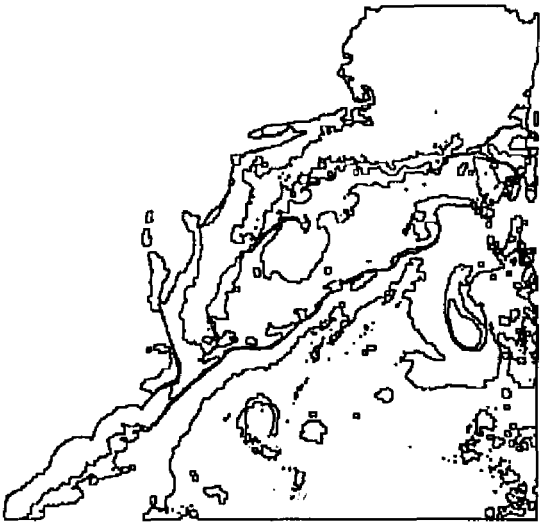
(a) original image



(b) edge image

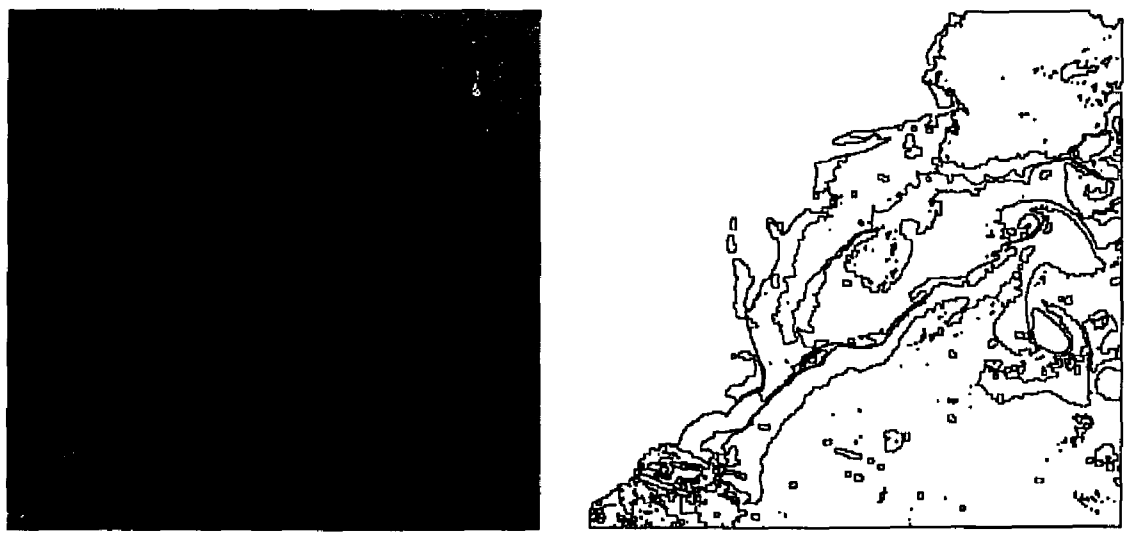


(c) original image

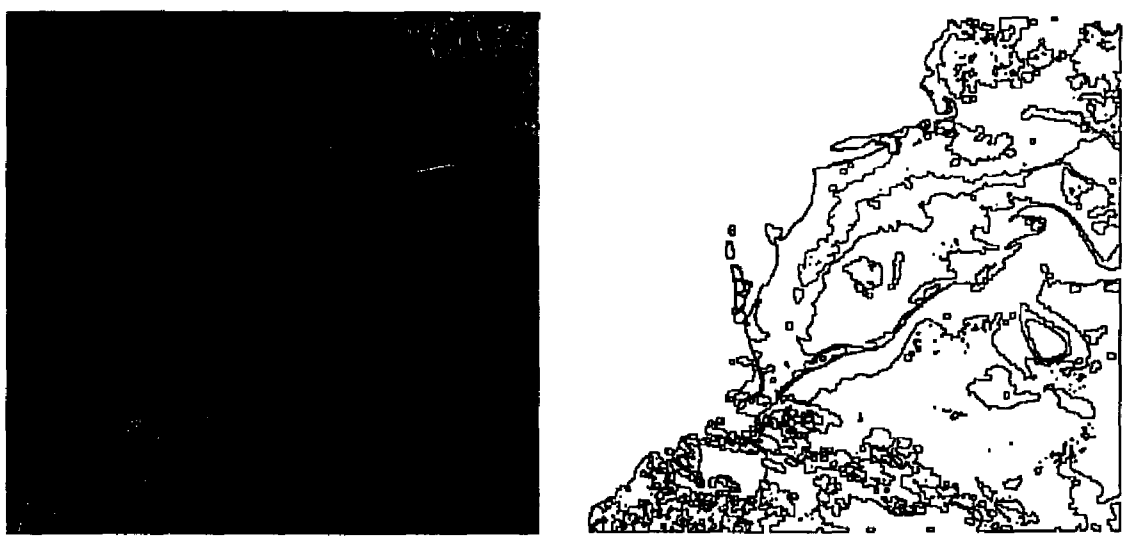


(d) edge image

Figure 5.10: Result of applying algorithm $E2$ version 2



(a) original image (b) edge image



(c) original image (d) edge image

Figure 5.11: Result of applying algorithm $E2$ version 2

-1	-2	-1
0	0	0
1	2	1

(a) S_h

-1	0	1
-2	0	2
-1	0	1

(b) S_v

Figure 5.12: The Sobel edge finding operators

show the result of thresholding the edge strength at 15 (the dynamic range of gray levels is $0 \ldots 255$). Notice the weakness of the response of the Sobel operator to the south edge of the gulf stream.

5.3.2 Morphological Methods

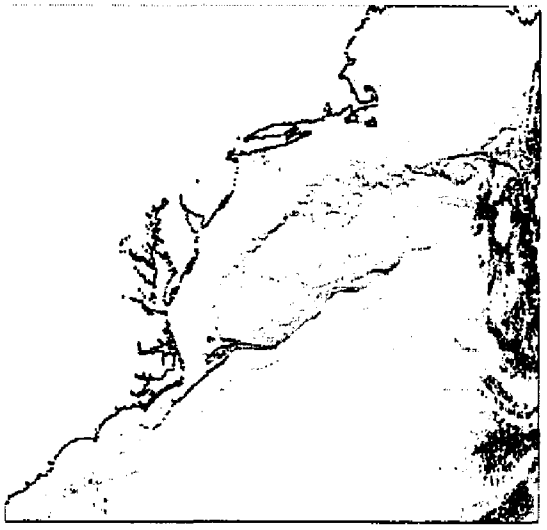
Another approach to edge detection involves a nonlinear method based on morphological filtering of an image. Details about image morphology can be found in [28]. In order to make this discussion more complete, we present a minimal description of morphology in the context of the computation of a morphological edge image.

The *dilation* of an image I by a morphological element M is defined as

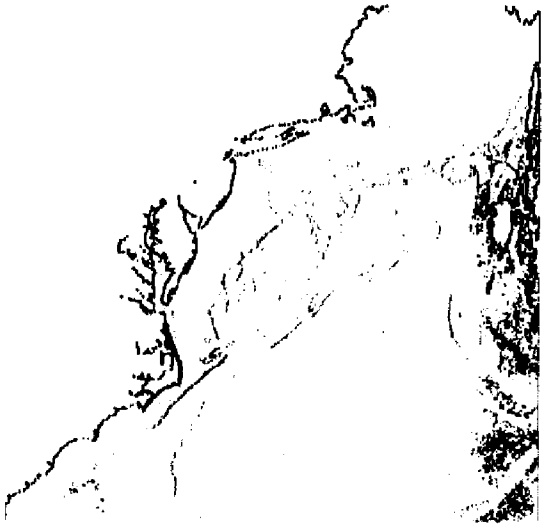
$$I \oplus M(u,v) = \max\{I(u-x, v-y)\} x, y \in D_M \tag{5.1}$$

where D_M is the domain of M. Informally, a dilation operation is the replacement of each pixel in the image by the maximum in a local neighborhood (centered around the pixel) determined by the geometry of the morphological element M. Analogously, the *erosion* of I by M is defined as

$$I \ominus M(u,v) = \min\{I(u-x, v-y)\} x, y \in D_M \tag{5.2}$$



(a) response to S_h



(b) response to S_v

(c) response (a) thresholded at 15

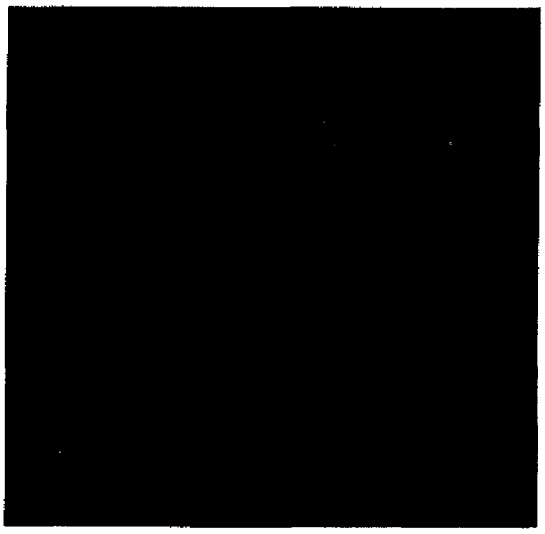
(d) response (b) thresholded at 15

Figure 5.13: Result of applying S_h and S_v to image in Figure 4.8(a)

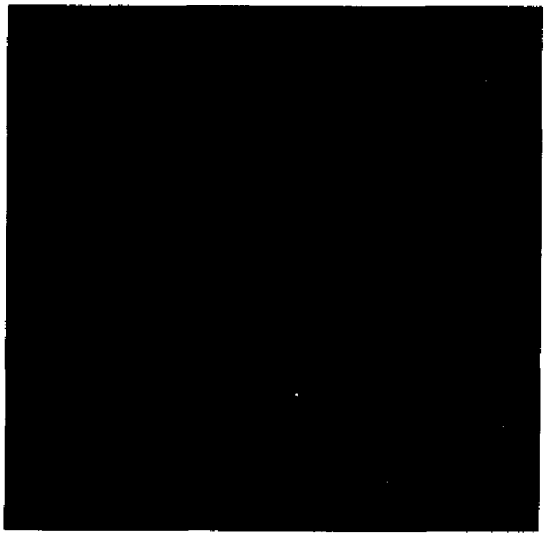
and this operation corresponds to picking the minimum gray level in the neighborhood of the pixel. The morphological gradient of an image, say G is given by

$$G = (I \oplus M) - (I \ominus M) \, . \tag{5.3}$$

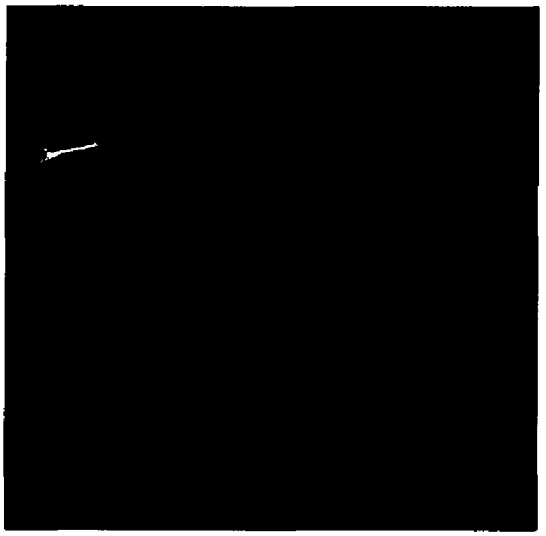
Figure 5.14 shows the result of morphological dilation, erosion and the morphological gradient of the image in (a).



(a) original image



(b) image (a) dilated



(c) image (a) eroded

(d) morphological gradient

Figure 5.14: Morphological operations on an image

Chapter 6

Spectrum

This chapter describes a portable, extensible toolkit for image processing based on conventional and multiresolution techniques. *Spectrum* was developed mainly to provide a testbed for verifying the theoretical methods proposed in this dissertation. Although several image processing software packages exist, both in the public domain and in the industry, they are usually quite large and complicated, especially if new algorithms and methods need to be incrementally added to the existing framework. Thus, there is an ever present need for a simple and extensible toolkit particularly suited for a research environment. *Spectrum* was designed and developed to fill this need.

Spectrum is an extensible, portable toolkit designed for development of image processing applications. Typical image processing tasks involve manipulation of existing image data and saving the results. *Spectrum* provides an efficient framework for computational research and development in image processing. It can be used either as a stand-alone tool or as a platform for building more complex image processing operations based on existing primitives.

Spectrum is portable. The lowest level modules (primitives) in *Spectrum* are designed to be platform-independent. Device-specific image processing functions such as those responsible for image visualization and interactive image manipulation are abstracted into cohesive modules which may be developed for any specific hardware platform, even one that is designed in the future. The underlying image

processing primitives remain the same; the interface between this kernel and the specific hardware platform to which the kernel is customized is modular and may be replaced or modified.

Spectrum encourages software reuse and the philosophy of 'mix and match' through a set of independent modules which may be combined in many ways to create solutions to new problems. It is also easy to incorporate a new algorithm, for example, into the system or to provide support for a new image file format. This modular design allows for easy customization to specific application domains and processing environments.

6.1 Introduction

As image processing evolves into a mature scientific discipline, a rich experimental sub-field for developing methodologies and comparative evaluation of different theories will emerge. A comprehensive set of tools tailored to image processing is both invaluable and necessary. Several research groups all over the world have developed 'toolkits' for image processing; some notable examples include the Utah Raster Toolkit and the Khoros System [36]. In the absence of standards, each group has had to develop its own method of storing and retrieving data. Many choices exist for image file formats – for instance compressed, uncompressed, run-length encoded, etc. Another notable feature about these packages is the fact that they are closely tied to a software operating environment. For example, Khoros modules are based on the MIT X11R4 and the Athena widget set.

Admittedly, it is not feasible to develop a software package or toolkit that is universally portable. Device and platform-specific considerations invariably creep

in at some level during the system design and development stage. A wider circle of compatibility is achievable if packages are designed to port across platforms. One possible method of accomplishing this is by tying the package to a high-level programming language rather than to a particular software environment. The package may then be designed in two parts – the device-independent part which is portable and a device-dependent part which is not portable across different environments. We will call the device-independent part of a package its *portable kernel* or just *kernel*, for short. There is an interesting contrast between our use of this term and the more usual interpretation – the kernel of the UNIX operating system is its device-dependent part !

In addition to the kernel, each application domain and processing environment needs device-specific modules for such operations as image display, data visualization, printing, interactive image manipulation etc. This part of any package is device-specific and we refer to this as the *shell* of the package. Thus each software package consists of a portable kernel and a device-specific shell. The basic idea is to structure the kernel so that most of the functionality of the package is encapsulated in the kernel. This enables the porting of the package, essentially unchanged, to various operating environments.

Spectrum has been designed according to the guidelines for portability outlined above. The kernel of *Spectrum* is built entirely in the 'C' programming language and is thus portable across several operating environments. In particular, the kernel is not tied to any operating system. Thus, the kernel source code may be compiled on any platform that supports a 'C' language compiler regardless of the software operating environment. The kernel consists of primitive image processing operations which may be cascaded together in one or more ways to accomplish complex image

processing tasks. In its current implementation, the device-specific shell of *Spectrum* was built using the *Graphics Library* of Silicon Graphics workstations.

In recent years, a lot of research interest has centered around multiresolution analysis and other wavelet-based methods of signal processing. Wavelet based methods offer a natural, hierarchical scheme of signal processing which is of particular use to images. The information in an image (scene) is quite different at different levels of image analysis. While other transform methods such as the Fourier Transform have become standard fare on many image processing tool kits, the Wavelet Transform has not received much attention in this area. *Spectrum* incorporates kernel modules which support both wavelet based image analysis and multiresolution techniques for conventional image processing tasks such as edge detection and segmentation.

6.2 System Overview

Figure 6.1 shows a schematic diagram of *Spectrum*. As described in the previous section, the kernel contains the portable part of the system and the shell contains the device-specific part. Image processing tasks appear in the kernel of the system and data visualization, and interactive image display and manipulation tasks appear in the shell of the system.

The circle represents the kernel of portable image processing operations. The rectangle around the circle represents the user interface between the kernel and the user.

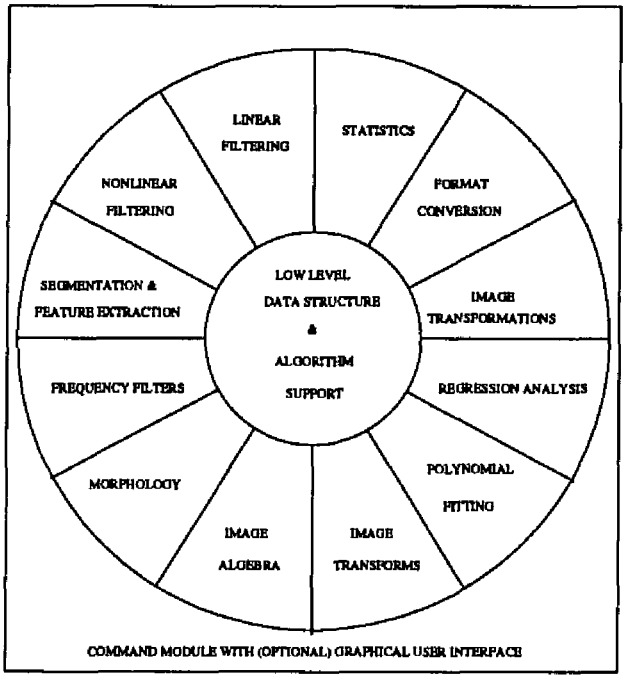


Figure 6.1: *Spectrum* : schematic diagram

6.3 System Kernel Components

The following sections contain a brief description of the functionality of the kernel of *Spectrum*. Each section describes one primitive in the kernel. Unless otherwise indicated, primitives accept RRLP binary format images as input and write RRLP binary format images as output. A more complete list may be found in [56].

adcon

adjusts the contrast in images. A set of control points, spanning the entire intensity level range, may be specified. Intensity levels in the input image are mapped to the specified output range using a piece-wise linear mapping between the control points. If no parameters are specified, the input image is just copied to the output.

`atorrl`

converts an ASCII format image to the RRLP binary format. ASCII images are common in satellite oceanography. AVHRR images are sometimes stored in ASCII form. Converting them to RRLP binary format saves space However, this format is not a compressed format.

`backfill`

implements a seedfill algorithm (see [24] for details) for segmenting the background in an image from the foreground. An optional seed value may be specified to start the algorithm. If no seed value is specified, a default value of BLACK is used as the seed value. The output of this The actual values of BLACK and WHITE depend on the dynamic range of intensity values allowed by the image file format module is an image with each pixel labeled BLACK or WHITE depending upon whether the pixel is classified as foreground or background. Textured images and images with no clear foreground and background yield unpredictable results.

`bmorph`

carries out binary morphological operations. The size of the morphological element and the type of operation are inputs to the module. *Dilation* and *Erosion* are the two supported functions. A dilation operation replaces every WHITE pixel element with a 'patch' of WHITE pixels, the patch size being determined by a parameter to the module. (see also gmorph)

`dwt`

computes the dyadic discrete wavelet transform of an image. The projection of the full resolution image onto the detail spaces is quantized separately from the projection onto the low resolution smooth space. For ease of viewing, the dynamic

range of the transform values (which are real numbers) is quantized to eight bits per pixel (between an integer value of 0 and 255, both inclusive).

`edget`

thresholds edge strength images; its use is limited in dealing with grayscale images since a simple threshold operation (see thresh) could just as well take care of thresholding edge strength images. Edge detection modules usually calculate the edge strength at each pixel in the input image and quantize the edge strength to the dynamic range of intensity levels allowed by the image file format.

Multi-band (multiple channel) images are typically analyzed for edges in the different data channels separately. These edge strength images may then be combined in several ways to arrive at the final output of the edge detector. Module *edget* implements various techniques for combining edge strength images from multiple channels. For instance, a logical operation between the edge strengths in the different channels may determine the final output. Alternatively, the minimum, the maximum or the average of the edge strengths in the different channels may define the final output.

`filter`

implements convolution-like filtering operations. Any filtering operation that can be modeled as a convolution can be carried out by invoking the module with appropriate filter-coefficient files. (see gauss, spotf and gmorph). This module buffers the entire image before output. In an image processing pipeline, this is likely to be the bottle neck.

A set of convolution kernels has been developed for use with this module. This set includes kernels for Gaussian convolution, averaging, the Sobel operator,

the Roberts operator, the sharpening filter, the Prewitt operator, and the digital Laplacian operator. This list is by no means exhaustive.

`gauss`

generates Gaussian and derivative of Gaussian convolution kernels for use with the filter module described earlier. The dimensions, mean and variance of the Gaussian may be specified as parameters.

`gmorph`

implements a generalization of the morphological operations described in the *morph* module earlier. Grayscale morphology is a relatively new area and it is not clear how multiple channels in images should be handled. Under the current implementation, *gmorph* replaces each pixel in the image with the maximum (minimum) intensity found in a small neighborhood of specified size centered around the pixel in question, for a dilation (erosion) operation.

`histogram`

calculates the histogram of an image. Several output modes are supported as also are several options for visualizing the histogram. The histogram may be of raw data or the data can be scaled according to a specified scale factor.

`invert`

carries out the inversion of an image with respect to an all-WHITE image (the RRLP binary format assumes a 'min-is-black' reference). That is, each pixel intensity value in the image is replaced by the value obtained by subtracting the intensity value from the maximum permissible intensity value.

`median`

carries out median rank filtering on an image. Typically, median filtering is carried out on grayscale images.

`mrht`

implements a multiresolution algorithm for analyzing image histograms. The image histogram is low-pass filtered and subsampled at different resolutions. The peaks and valleys detected in the subsampled histogram are used to obtain a partitioning of the dynamic range of intensity levels and a corresponding segmentation of the input image.

`pfit`

fits a bicubic interpolating polynomial over each neighborhood of specified size in the entire image. The first and second derivatives of this polynomial are then calculated and an edge strength image corresponding to the gradient is output. The gradient values found in the image are recoded (quantized) to span the dynamic range of allowed intensity levels.

`readppm`

reads a PPM format file and converts it into the RRLP format. It is typically used to read in a PPM file before the output of this module is streamed through to other *Spectrum* modules which accept only RRLP format images as input.

`readrrl`

reads in an RRLP format image file. The module is superfluous (nevertheless, it is provided for completeness) since any module that accepts as its input, an RRLP format file may be invoked by using the UNIX input redirecting operator '<'.

`rotate`

rotates an image clockwise and counter-clockwise. Since, the module always gives rectangular images as output, it is relevant to mention that the bounding box of any rotated image is larger than the image itself. The output image is cropped

to fit the rotated image. Rotations of multiples of $\pm\pi/2$ are implemented efficiently so that no trigonometric quantities are calculated.

`scale`

performs fast anamorphic and aspect-preserving scaling over images. Multiple image channels are supported. Images may be scaled up (enlarged) or down (shrunk) as specified.

`seedfill`

implements a seedfill algorithm described earlier (see backfill). The difference here is, foreground objects are segmented and labeled with unique numbers in the range of allowed intensity levels.

`spotf`

removes isolated rectangular spots of a specified size occurring in neighborhoods of a specified size. A default spot size of 1 pixel and a default neighborhood size of 5x5 are used if the corresponding parameters are unspecified.

`tfplot`

renders the time-frequency diagram of a one-dimensional signal as a two-dimensional image. The wavelet transform of the signal is computed and the coefficients are quantized to eight bits. This makes it possible to visualize the time-frequency diagram as a grayscale image where the brightness of the time-frequency rectangle is directly proportional to the magnitude of the associated wavelet coefficient.

`threshold`

opens a titled GL window on a display server and displays a 24-bit color image. Menu options for zooming in and out, histogram visualization, pixel intensity level view etc. are included.

`iedit`

displays a titled GL window and a subwindow with tools for modifying such parameters of images as brightness, saturation, sharpness, contrast, gamma values etc.

In summary, *Spectrum* is an easy-to-use toolkit for image processing. It has been designed and developed to be as platform-independent as possible. As a result, it is widely usable. Further, it has a modular structure which makes it easy to add functionality to *Spectrum* . With a minimum of coding effort, new modules can be incorporated into the existing framework to enhance its capabilities.

Chapter 7

Summary

This dissertation proposes and implements low-level image processing operations using ideas from the theory of wavelets and multiresolution. Multiresolution analysis provides a natural, hierarchical framework for analyzing and manipulating images and other signals and has wide applicability. The remainder of this chapter summarizes the specific contributions of the present dissertation and indicates possible future directions for research.

7.1 Contributions

The main contributions of this dissertation are two algorithms each for segmentation and edge detection in images. A secondary contribution is a portable toolkit for image analysis developed as a computational aid to test the algorithms proposed. For purposes of comparing these algorithms with others in the literature, it has been necessary to include other algorithms to this toolkit with the result it has grown into a full-fledged image processing toolkit. The following two sections summarize the segmentation and edge detection algorithms proposed; the toolkit is described in Chapter 6.

7.1.1 Segmentation

Two new algorithms $S1$ and $S2$ have been proposed, implemented and tested on various images. $S1$ uses several subsampled versions of a smoothed copy of an image

histogram to pick thresholds for segmenting the image. $S2$ employs a wavelet-based technique for projecting the original histogram onto successively finer subspaces of $L^2(\mathbf{R})$ to obtain increasingly refined segmentations of the image.

The performance of $S2$ is qualitatively better than that of $S1$ due to the fact that $S1$ uses a 'static' version of the histogram — once the original histogram is smoothed, all further computations are performed on the smoothed version; $S2$ on the other hand always works with the original histogram, and wavelet-smoothing this histogram to get its projections for threshold selection.

Both algorithms are robust in the sense that they are not sensitive to noise in the image since the smoothing operation protects their performance. The computational complexity of both techniques is linear in the size of the image — in fact, both make exactly two passes over the image and a small number of passes over the image histogram whose size depends only on the number of the levels of quantization used for encoding the image. Finally, both algorithms work with histograms which have multiple modes and/or high frequency components (rapidly varying pixel counts).

The performance of existing split-and-merge segmentation schemes can be improved by incorporating either of these algorithms in the 'split' phase of the segmentation algorithm.

7.1.2 Edge Detection

Two new edge detection algorithms $E1$ and $E2$ have been proposed, implemented and tested mainly on satellite images for evaluating their performance in the presence of weak edges and their tolerance to noise. Their results are compared with those of more traditional edge operators such as the digital Laplacian, the morphological edge operator [28], and the bicubic polynomial fitting method [27].

Algorithm $E1$ uses the wavelet coefficients in the detail spaces W_j of $L^2(\mathbf{R})$ to estimate the edge strength at pixels in the input image. Since, we are interested primarily in edge detection and isolation in satellite images which display weak gradients, $E1$ employs a logarithmic transformation on the wavelet coefficients before quantizing them for visualization.

Algorithm $E2$ uses the segmentation techniques developed in Chapter 4 to prepare an image before the application of an edge operator. This preprocessing operation enhances the noise-tolerance of the edge operator and enables the detection of weak edges. $E2$ is presented in two different versions. The second version of $E2$ is a minor modification of the first version; the second version has thinner edges.

7.2 Research Directions

A natural extension of this work is in the area of color image processing. The segmentation and edge detection ideas can be extended to operate on color images, and indeed, there is a lot of research in this area, currently in progress, all over the world.

On a different note, in recent years, parallel architectures and implementations are generating a lot of interest due to their increased computational efficiency. While many problems and areas in image processing have already been touched by this field, parallel techniques for multiresolution and wavelets are still in their nascent stage. Given that image processing ideally placed for parallel processing — most image processing operations involve the same computation done over a local neighborhood in the image repeated elsewhere in the image with different image

data — parallel multiresolution and wavelet techniques are potentially very useful in image processing.

There is a certain amount of 'inherent' parallelism in the idea of multiresolution processing since the different scales of resolution are decoupled, in a manner of speaking. Thus, a signal may be processed at various scales in parallel before the 'pieces are put back together.'

Edge detection and segmentation being fundamental and low-level problems in image processing, efficient parallel algorithms for these areas (with or without using wavelets and/or multiresolution) are always useful.

Bibliography

[1] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies. Image Coding using Vector Quantization in the Wavelet Transform Domain. In *Proc. 1990 IEEE Int. Conf. Acoust. Speech, Sig. Proc.*, Albuquerque. NM, Apr. 3–6, 1990.

[2] D. Ballard and C. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] J-M Beaulieu and M. Goldberg. Hierarchy in Picture Segmentation: A Stepwise Optimization Approach. *IEEE Trans. Pattern Anal. Machine Intell.*. **11**, 2, pp. 150–163, February 1989.

[4] J.J. Benedetto and M.W. Frazier Ed. *Wavelets: Mathematics and Applications*. CRC Press, 1994.

[5] G. Bongiovanni, L. Cinque, S. Levialdi and A. Rosenfeld. Image Segmentation by a Multiresolution Approach. *Pattern Recognition*. **26**, 12, pp. 1845–1854, 1993.

[6] P.J. Burt and E.H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. Commun.*. **COM-31**, 4, pp. 532–540, April 1983.

[7] P.L. Butzer, A. Fischer and K. Rückforth. Scaling Functions and Wavelets with Vanishing Moments. *Computers Math. Applic.*. **27**, 3, pp. 33–39, 1994.

[8] J. Canny. A Computation Approach to Edge Detection. *IEEE Trans. Pattern Anal. Machine Intell.*. **PAMI-8**, 6, pp. 679–698, November 1986.

[9] S-Y. Chen, W-C. Lin and C-T. Chen. Split-and-Merge Image Segmentation based on Localized Feature Analysis and Statistical Tests. *CVGIP: Graphical Models and Image Processing* **53**, 5, pp. 457–475, 1991.

[10] C.K. Chui. *An Introduction to Wavelets*. Academic Press, 1992.

[11] C.K. Chui Ed.. *Wavelets: A Tutorial in Theory and Applications*. Academic Press, 1992.

[12] I. Daubechies. Orthonormal Bases of Compactly Supported Wavelets. *Commun. Pure & Appl. Math.*. **41**, pp. 909–996, November 1988.

[13] I. Daubechies. *Ten Lectures on Wavelets*. Notes for the CBMS conference (Lowell), SIAM 1992.

[14] I. Daubechies. Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme. *SIAM J. Math. Anal.*. **24**, 2, pp. 499–519, March 1993.

[15] I. Daubechies. The Wavelet Transform, Time-Frequency Localization and Signal Analysis. *IEEE Trans. Info. Theory*. **36**, 5, pp. 961–1005, September 1990.

[16] J-P. Djamdji and A. Bijaoui. Goemetrical Registration of Images: The Multiresolution Approach. *Photogrammetric Engineering and Remote Sensing*. **59**, 5, pp. 645-653, May 1993.

[17] D.E. Elliot and K.R. Rao. *Fast Transforms: Algorithms, Analyses, Applications*. Academic Press, 1982.

[18] D. Esteban and C. Galand. Applications of Quadrature Mirror Filters to Split Band Voice Coding Schemes. In *Proc. Int. Conf. Acoust., Speech & Signal Proc.*, May 1977.

[19] J. Froment and S. Mallat. Second Generation Compact Image Coding with Wavelets. *Wavelets: A Tutorial in Theory and Applications*. pp. 655–678, Academic Press, 1992.

[20] J.M. Gauch and S.M. Pizer. Multiresolution Analysis of Ridges and Valleys in Grey-Scale Images. *IEEE Trans. Pattern Anal. Machine Intell.*. **PAMI-15**, 6, pp. 635–646, June 1993.

[21] R.C. Gonzalez and P. Wintz. *Digital Image Processing*. 2nd Ed., Addison-Wesley 1987.

[22] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison-Wesley 1992.

[23] R.A. Gopinath and C.S. Burrus. Efficient Computation of Wavelet Transforms. In *Proc. 1990 IEEE Int. Conf. Acoust. Speech, Sig. Proc.*, Albuquerque. NM, Apr. 3–6, 1990.

[24] A. Glassner, J. Arvo and D. Kirk, Eds.. *Graphics Gems*. Academic Press, 1991.

[25] A. Haar. Zur theorie der Orthogonalen Funktionensysteme. *Math. Ann.*. **69**, pp. 331-371, 1910.

[26] J.F. Haddon and J.F. Boyce. Image Segmentation by Unifying Region and Boundary Information. *IEEE Trans. Pattern Anal. Machine Intell.*. **12**, 10, pp. 929–948, October 1990.

[27] R.M. Haralick. Digital Step Edges from Zero Crossing of Second Directional Derivatives. *IEEE Trans. Pattern Anal. Machine Intell.*. **6**, 1, pp. 58–68, January 1984.

[28] R.M. Haralick, S.R. Sternberg and X. Zhuang. Image Analysis using Mathematical Morphology. *IEEE Trans. Pattern Anal. Machine Intell.*. **9**, 4, pp. 58–68, July 1987.

[29] R.M. Haralick and L.G. Shapiro. Survey: Image Segmentation Techniques. *Comput. Vision Graphics, Image Proc.*. **29**, pp. 100-132, 1985.

[30] C.E. Heil and D.F. Walnut. Continuous and Discrete Wavelet Transforms. *SIAM Review*. **31**, 4, pp. 628–666, December 1989.

[31] C. Herley and M. Vetterli. Wavelets and Recursive Filter Banks. *IEEE Trans. Signal Proc.*. **41**, 8, pp. 2536–2556, August 1993.

[32] C. Herley and M. Vetterli. Wavelets and Recursive Filter Banks. *IEEE Trans. Signal Proc.*. **42**, 10, pp. 2650–2663, October 1994.

[33] R.J. Holyer and S. Peckinpaugh. Edge Detection Applied to Satellite Imagery of the Oceans. *IEEE Trans. Geosc., Rem. Sens.*. **27**, 1, pp. 46–56, January 1989.

[34] A.K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[35] B. Jawerth and W. Sweldens. An Overview of Wavelet Based Multiresolution Analyses. *Siam Rev.*. **36**, 3, pp. 377–412, September 1994.

[36] K. Konstantinides and J.R. Rasure. The Khoros Software Development Environment for Image and Signal Processing. *IEEE Trans. Image Proc.*. **3**, 3, pp. 243–252, May 1994.

[37] J-S. Lee, Y-N. Sun, C-H. Chen and C-T. Tsai. Wavelet based Corner Detection. *Pattern Recognition*. **26**, 6, pp. 853–865, 1993.

[38] L.M. Lifshitz and S.M. Pizer. A Multiresolution Hierarchical Approach to Image Segmentation based on Intensity Extrema. *IEEE Trans. Pattern Anal. Machine Intell.*. **PAMI-12**, 6, pp. 529–540, June 1990.

[39] S.A. Little, P.H. Carter and D.K. Smith. Wavelet Analysis of Bathymetric Profile reveals Anomalous Crust. *Geophysical Research Letters*. **20**, 18, pp. 1915–1918, September 1993.

[40] J. Liu and Y. Yang. Multiresolution Color Image Segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*. **PAMI-16**, 7, pp. 689–700, July 1994.

[41] S. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans. Pattern Anal. Machine Intel.*. **PAMI-11**, 7, pp. 674–693, July 1989.

[42] S. Mallat. Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(R)$. *Trans. Amer. Math. Soc.*. **315**, pp. 69–88, 1989.

[43] S. Mallat. Multifrequency Channel Decompositions of Images. *IEEE Trans. Acous., Speech & Signal Proc.*. **ASSP-37**, 12, pp. 2091–2110, December 1989.

[44] S. Mallat and W.L. Hwang. Singularity Detection and Processing with Wavelets. *IEEE Trans. Info. Theory*. **38**, 2, pp. 617–643, March 1992.

[45] S. Mallat and Z. Zhang. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Trans. Signal Proc.* **41**, 12, pp. 3397–3415, December 1993.

[46] S. Mallat and S. Zhong. Characterization of Signals from Multiscale Edges. *IEEE Trans. Pattern Anal. Machine Intel.*. **PAMI-14**, 7, pp. 710–732, July 1992.

[47] D. Marr and E. Hildreth. Theory of Edge Detection. *Proc. Roy. Soc. Lond.*. **207**, pp. 187-217, 1980.

[48] Y. Myer. *Wavelets and Operators*. Cambridge University Press, 1992.

[49] Y. Myer. *Wavelets: Algorithms and Applications*. SIAM, Philadelphia, 1993.

[50] V.S. Nalwa and T.O. Binford. On Detecting Edges. *IEEE Trans. on PAMI*. **PAMI-8**, 6, pp. 699–714, November 1986.

[51] L. Prasad, S.S. Iyengar, R.L. Rao and R.L. Kashyap. Fault-Tolerant Sensor Integration Using Multiresolution Decomposition. *Physical Review E.* **49**, 4, pp. 3452–3461, April 1994.

[52] T.N. Pappas. An Adaptive Clustering Algorithm for Image Segmentation. *IEEE Trans. Signal Proc.*. **40**, 4, pp. 901–914, April 1992.

[53] L. Prasad, S.S. Iyengar, R.L. Rao and R.L. Kashyap. Fault-Tolerant Integration of Abstract Sensor Estimates Using Multiresolution Decomposition. In *Proc. IEEE Conf. Sys., Man & Cyber.*. Le Touquet, France, Oct. 1993

[54] T.Y. Phillips, A. Rosenfeld and C.A. Sher. $O(\log n)$ Bimodality Analysis. *Pattern Recognition*. **22**, pp. 741–746, 1989.

[55] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery. *Numerical Recipes in C*. Second Edition, Cambridge University Press, 1992.

[56] R.L. Rao. *Spectrum* : A Portable, Extensible Toolkit for Conventional & Multiresolution Image Processing. *Technical Report*, Robotics Research Laboratory, Louisiana State University, 1994.

[57] O. Rioul and P. Duhamel. Regular Wavelets: A Discrete-Time Approach. *IEEE Trans. Signal Proc.*. **41**, 12, pp. 3572–3579, December 1993.

[58] O. Rioul and P. Duhamel. Fast Algorithms for Discrete and Continuous Wavelet Transforms. *IEEE Trans. Info. Theory*. **38**, 2, pp. 569–586, March 1992.

[59] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Proc. Mag.*. pp. 14–38, 1991.

[60] A.A. Rodriguez and O.R. Mitchell. Image Segmentation by successive Background Extraction. *Pattern Recognition*. **24**, 5, pp. 409–420, 1991.

[61] A. Rosenfeld, Ed.. *Multiresolution Image Processing and Analysis*. Springer-Verlag, New York, 1982.

[62] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*. Volume 2, 2nd Edition. Academic Press, 1982.

[63] A. Rosenfeld and M. Thurston. Edge and Curve Detection for Visual Scene Analysis. *IEEE Trans. Computing*. **C-20**, pp. 562–569, 1971.

[64] R.J. Schalkoff. *Digital Image Processing and Computer Vision*. John Wiley and Sons Inc. 1989.

[65] L.L. Schumaker and G. Webb, Ed.. *Recent Advances in Wavelet Analysis*. Academic Press, 1994.

[66] G. Strang. Wavelets. *American Scientist*. **82**, pp. 250–255, May-June 1994.

[67] G. Strang. Wavelet Transforms versus Fourier Transforms. *Bull. Am. Math. Society*. **28**, pp. 288–305, 1993.

[68] R.S. Strichartz. How to make Wavelets. *Amer. Math. Monthly*. **100**, 5, pp. 539–556, June 1993.

[69] S.L. Tanimoto and T. Pavlidis. A Hierarchical Data Structure for Image Processing. *Comput. Graphics & Image Proc.*. **4**, pp. 104–119, 1975.

[70] J.G. Teti and H.N. Kritikos. SAR Ocean Image Representation using Wavelets. *IEEE Trans. Geosc. and Remote Sensing*. **30**, 5, pp. 1089–1094, September 1992.

[71] A. Teuner and B.J. Hosticka. Adaptive Gabor Transformation for Image Processing. *IEEE Trans. Image Proc.*. **2**, 1, pp. 112–117, January 1993.

[72] V. Torre and T. Poggio. On Edge Detection. *IEEE Trans. Pattern Anal. Machine Intell.*. **PAMI-8**, 2, pp. 147–163, March 1986.

[73] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ.

[74] P.P. Vaidyanathan. Multirate Digital Filters, Filter Banks, Polyphase Networks and Applications: A Tutorial. *Proc. IEEE*. **78**, pp. 56–93, January 1990.

[75] M. Vetterli. A Theory of Multirate Filter Banks. *IEEE Trans. Acoust., Speech and Signal Proc.*. **ASSP-35**, 3, pp. 356–372, March 1987.

[76] M. Vetterli and D. Le Gall. Perfect Reconstruction FIR Filter Banks: Some Properties and Factorizations. *IEEE Trans. Acoust., Speech and Signal Proc.*. **37**, 7, pp. 1057–1071, July 1989.

[77] M. Vetterli and C. Herley. Wavelets and Filter Banks: Theory and Design. *IEEE Trans. Signal Proc.*. **40**, 9, pp. 2207–2231, September 1992.

[78] G.G. Walter. A Sampling Theorem for Wavelet Subspaces. *IEEE Trans. Informat. Theory*. **38**, 2, pp. 881–884, March 1992.

[79] G.G. Walter. *Wavelets and Other Orthogonal Systems*. CRC Press, 1994.

[80] M.V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A.K. Peters, 1994.

[81] R. Wilson, A.D. Calway and E.R.S. Pearson. A Generalized Wavelet Transform for Fourier.... *IEEE Trans. Informat. Theory*. **38**, 2, pp. 674–690, March 1992.

[82] S. Wolfram. *Mathematica: A System of Doing Mathematics by Computer*. 2nd Ed., Addison-Wesley, 1991.

[83] X. Wu. Adaptive Split-and-Merge Segmentation Based on Piecewise Least-Square Approximation. *IEEE Trans. Pattern Anal. Machine Intell.*. **15**, 8, pp. 808–815, August 1993.

[84] X. Xia and Z. Zhang. On Sampling Theorem, Wavelets and Wavelet Transforms. *IEEE Trans. Signal Proc.*. **41**, 12, pp. 3524–3535, December 1993.

Vita

Ramana Rao received a bachelor's degree in electrical engineering from the Indian Institute of Technology, Madras, India in 1988. He received a master's degree in systems science from the Department of Computer Science, Louisiana State University, Baton Rouge, Louisiana in 1994. He is currently a graduate research assistant in the Theoretical Division of the Los Alamos National Laboratory, Los Alamos, New Mexico. His research interests include image processing, wavelets, multiresolution, combinatorial optimization and computational geometry.

DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Ramana L. Rao

Major Field: Computer Science

Title of Dissertation: Multiresolution Techniques in Image Processing

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

04/03/1995