To the Graduate Council:

I am submitting herewith a thesis written by Osama Neiroukh entitled "RANGE IMAGE SEGMENTATION THROUGH MULTIRESOLUTION ANALYSIS USING WAVELETS." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

M. A. Abidi

M. A. Abidi, Major Professor

We have read this thesis
and recommend its acceptance:

Marshall Pace

Daniel B. Koch

Accepted for the Council:

C W Minkel

Associate Vice Chancellor
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature OSAMA NEROUKH

Date 3/14/1995

RANGE IMAGE SEGMENTATION THROUGH MULTIRESOLUTION ANALYSIS USING WAVELETS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Osama Neiroukh

May 1995

ACKNOWLEDGMENTS

I wish to extend sincere appreciation to my advisor Prof. Mongi Abidi, without whom this work would not have been possible. I also received help from several individuals, a special mention goes to Dr. Hamed Sari-Sarraf, Mr. Chandra Tan, and Mr. Mohamed Baccar for the numerous suggestions and hints along the way. Thanks are due to Prof. M. O. Pace and Prof. Koch for reading this thesis and being part of my defense committee. I would also like to thank Katie Jager for her help in reviewing and editing the manuscript.

ABSTRACT

Range images are important in robotic environments as they provide us with three–dimensional depth information about a scene. However, this information is too low–level for performing scene interpretation directly. Several computational steps are needed in order to extract and characterize meaningful features from raw range data. One of these computational steps is segmentation, the process of partitioning the image into groups of pixels that reveal the inner borders between three–dimensional scene elements and surfaces in an image.

In this research, a segmentation technique for range images is proposed using the wavelet transform (WT) and multiresolution analysis. The wavelet transform is used as a smoothing multiscale differentiation operator. A model for range image features is developed based on the differential properties of step and roof edges in range images. By applying one–dimensional oriented band pass filters at multiple scales, use of the wavelet transform avoids the execution delays associated with area–based segmentation algorithms for range image analysis. The result is a segmented image with labeled regions indicating different surface patches.

The developed algorithm is reasonably fast, taking under three seconds for the complete segmentation of a 128 × 128 range image on a Sun SPARCstation. The computations are implemented efficiently and may be performed in parallel. In computing the transform, one automatically has access to image features of various sizes that are enhanced and available for three–dimensional reconstruction and recognition tasks.

TABLE OF CONTENTS

CHAPTER **PAGE**

1. Introduction **1**

1.1 Problem Scope & Definition 1

1.2 Overview of Related Work 4

1.2.1 Range Image Analysis 5

1.2.2 Wavelet Transform and Multiresolution Analysis 8

1.3 Technical Notations 11

2. Time–Frequency Analysis and the Continuous Wavelet Transform (CWT) **12**

2.1 The Fourier Transform (FT) 13

2.2 The Gabor Transform, the Short–Time Fourier Transform (STFT), and the Uncertainty Principle 14

2.3 The Continuous Wavelet Transform (CWT) 15

2.3.1 Definition 16

2.3.2 The CWT in Higher Dimensions 17

2.4 Multiresolution Analysis 18

2.4.1 The Closed Subspaces V_j and Scaling Functions 18

2.4.2 The Detail Spaces W_j and Wavelet Functions 21

3. The Discrete Wavelet Transform (DWT) **24**

CHAPTER PAGE

3.1 Mallat's Multiresolution Algorithm 24

3.1.1 A Multiresolution Transform 24

3.1.2 Implementation of a Multiresolution Transform 26

3.1.3 The Orthogonal Wavelet Representation 28

3.1.4 Implementation of an Orthogonal Wavelet Representation 30

3.1.5 Extension of Wavelet Representations to Images 31

3.2 The *À Trous* Algorithm 36

3.2.1 The Decimated *À Trous* Algorithm 36

3.2.2 The Undecimated *À Trous* Algorithm 40

3.3 Comparison and Applications to Image Processing 41

4. Wavelets for Image Processing **43**

4.1 Properties of Wavelets 44

4.1.1 Orthogonality 44

4.1.2 Support 44

4.1.3 Symmetry 45

4.1.4 Regularity 46

4.1.5 Vanishing Moments 46

4.2 Examples 48

4.2.1 The Haar Wavelets 48

4.2.2 The Shannon Wavelets 50

4.3 Choice of Wavelets for Feature Extraction in Range Images 52

4.3.1 Desirable Properties of Candidate Wavelets 52

CHAPTER PAGE

4.3.2 Proposed Wavelet Family 54

5. Derivative Estimation for Range Images Using Wavelets . . . 63

5.1 Multiscale Derivative Approximations for 1–D Signals 64

5.2 Extension to Images 65

5.2.1 2–D Directional Derivatives Approximations 65

5.2.2 Magnitude and Angle of 2–D Wavelet Transform 67

5.3 Modulus Maxima of the Wavelet Transform 67

5.3.1 Maxima and Zero–Crossing Detection for Edges 68

5.3.2 Step Edge Detection 71

5.3.3 Roof Edge Detection 72

5.4 Range Images and Angle of 2–D Wavelet Transform 73

5.5 Practical Considerations 74

6. Experimental Results 76

6.1 Wavelet Transform Modulus Maxima Tracing Algorithm 77

6.1.1 Preprocessing 77

6.1.2 The *À Trous* Module 81

6.1.3 Modulus Maxima Tracing 81

6.1.4 Experimental Results 82

6.1.5 Analysis of Results 87

6.2 Wavelet Transform Angle Algorithm 89

6.2.1 Analysis of Results 91

6.3 Noise Performance of the Wavelet Transform Angle Algorithm . . 100

CHAPTER **PAGE**

6.4 Time Analysis for the Wavelet Transform Angle Algorithm 102

7. Summary and Conclusions **106**

7.1 Summary 106

7.2 Contributions 107

BIBLIOGRAPHY **109**

VITA **115**

LIST OF FIGURES

FIGURE		PAGE
1.1	*Illustration of intensity versus range images for the real image RECYCLEBOX.*	2
2.1	*Comparison of Gabor and Wavelet transform time–frequency resolution cells.*	15
2.2	*Projection of a function $f(x)$ onto the Haar spaces V_{-1} and V_0.*	19
3.1	*A single stage in Mallat's multiresolution algorithm [40].*	27
3.2	*A stage in Mallat's multiresolution algorithm for images.*	34
3.3	*The 2–D DWT: (a) Schematic representation of the 2–D DWT, (b) Original image, (c) Wavelet decomposition into three multiresolution layers, and (d) Approximation of image at three resolutions.*	35
3.4	*Dilation of a function using an interpolating filter*	38
3.5	*The undecimated DWT.*	41
4.1	*Examples of functions in the subspaces V_i.*	49
4.2	*The Haar wavelets: (a) Haar scaling function, (b) Haar wavelet function, (c) Fourier transform of Haar scaling function, and (d) Fourier transform of Haar wavelet function.*	51
4.3	*Shannon wavelets: (a) Shannon scaling function, (b) Shannon wavelet function, (c) Fourier transform of Shannon scaling function, and (d) Fourier transform of Shannon wavelet function.*	53

FIGURE PAGE

4.4 *Characterization of four–coefficient scaling functions: The* (c_0, c_3) *plane showing the ellipse* $c_0^2 + c_3^2 - c_0 - c_3 + c_0c_3 = 1$*, the circle* $(c_0 - 1/2)^2 + (c_3 + 1/2)^2 = \sqrt{2}/2$*, the line* $c_0 + c_3 = 1/2$*, and a triangle. The crosses at the intersection of the line and circle are the Daubechies* D_4 *scaling function [15].* 58

4.5 *Chosen four–coefficient symmetric scaling functions and their associated wavelets: (a) and (b):* $c_0 = c_3 = 0.05$*, (c) and (d):* $c_0 = c_3 = 0.10$*, (e) and (f):* $c_0 = c_3 = 0.20$*, and (g) and (h):* $c_0 = c_3 = 0.25$. 61

4.6 *Magnitude response (dB) versus normalized frequency of chosen scaling functions (solid line), wavelets (dotted line), and all–pass filters (dashed line) for the cases (a):* $c_0 = c_3 = 0.05$*, (b):* $c_0 = c_3 = 0.10$*, (c):* $c_0 = c_3 = 0.20$*, and (d):* $c_0 = c_3 = 0.25$. 62

5.1 *The 2–D DWT of a synthetic image.* 69

5.2 *The 2–D DWT of a real image.* 70

5.3 *A typical step edge and its first two derivatives.* 71

5.4 *Example of roof edges.* 72

5.5 *Profile of a sample roof edge and its first two derivatives.* 73

6.1 *Block diagram for the WT modulus maxima tracing algorithm.* . . 77

6.2 *Real range image TOOLS. The image was reduced from* 300×300 *to* 128×128 *pixels. (Original image provided by Coleman Research Corporation.)* 79

FIGURE **PAGE**

6.3 *Real range image TOOLS before (top) and after (bottom) geometric filtering.* 80

6.4 *Tracing of wavelet maxima using angle image.* 83

6.5 *Segmentation of real range image TOOLS using wavelet maxima.* 84

6.6 *Final edge map for image TOOLS at scales* 0 *to* 5. 85

6.7 *Segmentation of synthetic range image VW using wavelet maxima.* 85

6.8 *Segmentation of real range image CUP using wavelet maxima.* . . 86

6.9 *Segmentation of real range image PIPES using wavelet maxima.* . 86

6.10 *Final edge map for image PIPES at scales* 0 *to* 5. 87

6.11 *Block diagram for modulus maxima of angle algorithm.* 90

6.12 *A 3-D projection of synthetic range image CURVBLOCK. (Original range image obtained from the Michigan State/Washington State Universities Range Image Database.)* . . . 92

6.13 *Segmentation of image CURVBLOCK showing intermediate results.* 93

6.14 *Segmentation of synthetic range image ROOFS.* 94

6.15 *Segmentation of synthetic range image ESCHERKNOT.* 95

6.16 *Segmentation of real range image RECYCLEBOX.* 96

6.17 *Segmentation of real range image OCCLUSION.* 97

6.18 *Effect of noise on segmentation of real range image RECYCLEBOX, showing the images used and the number of regions in the final segmentation.* 101

CHAPTER 1

Introduction

Range data is often cited as an important source of three–dimensional (3–D) information about a scene as it gives us a depth map of the scene. Range can be defined as the distance of objects in the incident scene as measured by the range–sensor from a known point of origin. Several types of sensors can be used for gathering range data about a scene, including laser, infrared, and radar sensors. Active laser range scanners are the most widely used systems in range image acquisition. The technology used in laser scanners has improved significantly in the last few years, allowing for accurate measurements to be taken. Successful analysis of these range images has a vast potential for autonomous robot navigation, scene mapping, and tool manipulation tasks. The focus in this work is on the segmentation of range images, with a view to 3–D feature extraction and recognition.

1.1 Problem Scope & Definition

In order for a range image to be usefully analyzed, several algorithms are needed to transform the raw data obtained from a range scanner to bring out the desired features in a given image. Preprocessing raw range images includes

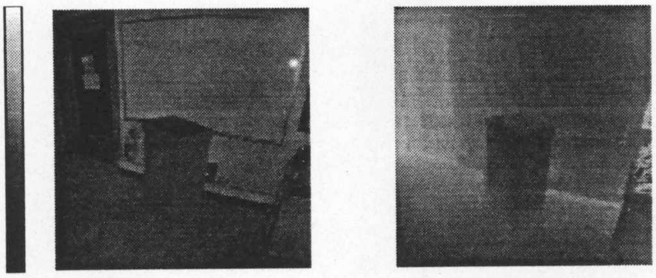


Figure 1.1: *Illustration of intensity versus range images for the real image RECYCLEBOX.*

clipping the image to the desired region of distance measurements, filtering any noise present in the data and correcting for known visual distortions such as the elliptical coordinates effect associated with point scanners, otherwise known as the cosine effect. Once this has been accomplished, the range image is segmented by dividing it into two–dimensional (2–D) primitive surfaces. The term edge detection is often used for this operation. The focus in range image analysis is on extracting different surface orientations that exist *within* an object as well as boundaries between different objects. Jumps in range values are labeled as step edges, and represent two surfaces such as an object and its background or occluding surfaces. All other types of edges found in range data are collectively referred to as roof edges. A range image illustrating step and roof edges is shown in Fig. 1.1. Step edges occur at the boundary of the box, whereas roof edges separate the different sides within the box. Detection and classification of roof edges is necessary for 3–D feature recognition and reconstruction of a scene.

Edge detection algorithms can be broadly divided into two types, edge–based and region–based methods [26]. Edge–based methods attempt to find the edges in the scene by differentiation of the image and classifying them accordingly. Area–based algorithms grow areas from initial *seeds* which are expanded or grown until a surface boundary is encountered. The resulting dividing lines between these areas are classified into different edge types. Area–based algorithms tend to be more accurate than their edge–based counterparts but present a computational challenge even for today's fast computers. They also require a starting point generated by an edge–based or erosion algorithm or other human intervention to aid in the selection and placement of seeds. Edge–based algorithms execute much faster and can give preliminary segmentations that can be refined if needed using an area–based algorithm.

This work fits under the category of edge–based segmentation. We use the wavelet transform (WT) to approximate derivatives of discrete data for analysis and segmentation of range images. A set of wavelets, to be used specifically for derivative estimation and edge localization in images, are derived and shown applicable to the segmentation of range images. The most commonly used methods for calculating derivatives at a pixel in a digital image apply a convolution mask or a set of masks to that pixel. Other methods fit a 2–D polynomial to a window centered around that pixel and calculate derivatives from the coefficients of these polynomials. The discrete wavelet transform (DWT) is based on one–dimensional (1–D) convolutions applied to an image as directional filters. This constitutes a generalization of the convolution–based operators, since any of them can be derived as a special case of the DWT.

This thesis is organized as follows. The rest of this chapter presents a literature review of related material. Chapter 2 introduces the continuous wavelet transform (CWT) and the concepts of multiresolution analysis. Chapter 3 describes different methods of discretizing the WT, with focus on Mallat's implementation [38, 40] and the *à trous* implementation [20, 50]. Chapter 4 discusses wavelets and their mathematical properties as they relate to range image modeling and processing. It covers also the conditions that are necessary for a wavelet to be used for feature extraction and edge detection. Chapter 5 relates derivative estimation for discrete signals to the DWT. Chapter 6 gives two algorithms that were developed based on the DWT for the segmentation of range images. Chapter 6 also presents extensive experimental results for sets of synthetic and real images. Finally, Chapter 7 discusses the results and gives some conclusions.

1.2 Overview of Related Work

As has been pointed out, this research investigates the use of wavelet analysis for some of the problems associated with range image analysis. This is the first attempt that the author is aware of that applies wavelet analysis to range data for the purposes of segmentation. A broad spectrum of literature was reviewed for the purposes of this research, covering both topics separately. Some of the more recent publications are covered below.

1.2.1 Range Image Analysis

Inokuchi *et al.* [33] present an edge/region segmentation ring operator for range images. The ring operator extracts a 1–D periodic function of range values that surround a given pixel. This function is transformed to the frequency domain using a fast Fourier transform (FFT) algorithm. Planar–region, step–edge, convex–roof edge, and concave–roof edge pixels are distinguished by examining the 0th, 1st, 2nd, and 3rd frequency components of the ring surrounding that pixel. These pixel types are grouped together and the resulting regions and edges are labeled. The ring–operator is reported to compute roof edges fairly well at planar surface boundaries on high quality data.

Mitiche and Aggarwal [45] develop a noise–insensitive edge detector based on a probabilistic model that accounts for range measurements errors. The computational procedure can be summarized as follows. (1) Step edges are extracted first from a range image using standard techniques. (2) For each discrete image direction (of four possible directions) at each pixel, a roof edge is hypothesized. For each hypothetical roof edge, two planes are fitted to the immediate neighborhood of the pixel, and the dihedral angles between these planes are computed. (3) Pixels are disregarded if all dihedral angles are less than a threshold. Bayesian likelihood ratios are computed to find the most likely edge direction. (4) Remaining pixels are passed through a non–maxima suppression algorithm that leaves only the desired edge pixels.

Oshima and Shirai [46] use a region–based segmentation for describing a range scene. Their image segmentation consists of several steps, including (1) grouping the image points into planar surface elements, (2) merging these surface elements into elementary regions, (3) classifying the regions into planar and curved regions, (4) merging similar adjacent regions into larger global regions and fitting quadrics to them, and (5) describing the complete scene in terms of the properties of global regions and the relations between these regions.

Besl and Jain [5, 6] introduce a least squares technique for fitting polynomials to pixel windows and approximated first and second derivatives from the fitted polynomials. They use these derivatives to classify each pixel in the image according to curvature signs computed from the derivatives. Once an initial labeling is obtained, they use an iterative region-growing method based on variable-order polynomial surface fitting to refine the segmentation. Besl authored a book [4] describing a complete system for range image understanding.

Haralick [54] introduces facet models for representing an image and then combines area-growing and edge detection into one process. Two facet models are experimented with and shown to converge for different sets of data. In [30], a directional derivative zero-crossings edge operator is introduced for the detection of digital step edges. The performance is compared with some of the classic edge operators, the generalized Prewitt gradient operator, and the Marr-Hildreth operator [43].

Taylor *et al.* [55] present a fast technique for segmenting surfaces in range images into planar regions. Their approach relies on a *split-and-merge* strategy. The criteria used is based on three parameters; the range value and two angles describing the orientation of the normal to the local best fit plane. In most cases, a fast comparison between the two angles provides the basis for splitting/merging possibilities. In cases where this comparison is inconclusive, a more complex range continuity test is performed using the original range value.

Parvin and Medioni [47] extract features such as edges, creases, and ridge lines from range images by fitting continuous functions to variable supports of data points. They use a rich set of primitives for describing the features found. Their method works at multiple scales, but they designed a control strategy to automatically select the most appropriate mask size. This mask is then used for calculating the best estimate of the zero-crossings of the first and second derivatives for the computed polynomials. They circumvent the correspondence problem between features at different scales by locally finding the best scale to describe a given feature.

Yokoa and Levine [56] propose using the signs of Gaussian and mean curvatures for characterizing local surface properties of smooth surfaces. These curvatures are computed by fitting biquadratic forms to a 5×5 window and using the fitted forms to estimate partial derivatives. From these partial derivatives, surface normal and curvatures are computed. Surface information is then fused with step and roof edge maps to produce a more reliable final segmentation.

Fan [21] infers surface features for segmentation using the zero-crossings and extremal values of the surface curvature measures. The detected features are used to segment a complex surface into simpler, meaningful components called surface patches. A multiview representation is introduced for each 3-D object model. These views are 2-D or $2\frac{1}{2}$-D descriptions and represent one particular view of the object model. The primitives of the system are attributed graphs. A description of a 3–D surface includes graphs whose nodes contain the information about the surface patches and the links represent the relationships between the surface patches.

Baccar [2] develops a Gaussian weighted least squares (GWLS) technique for polynomial fitting and applies it to range images. He uses weighted sets of orthogonal polynomials to maintain computational efficiency, and develops two segmentation algorithms based on GWLS techniques. He also investigates the characterization of arbitrarily shaped regions in range images and applies these techniques to synthetic and real range images.

1.2.2 Wavelet Transform and Multiresolution Analysis

Use of the WT has rapidly gained popularity in the last few years as a powerful tool in the multiscale analysis of discrete data. Although the idea of using different scales for signal analysis was first proposed in 1910 [29], these techniques were not fully developed or utilized until the work done by the French scientists Morlet, Grossman, and Meyer under the name *ondelettes* or wavelets [27, 28, 35, 44]. Subsequently, Daubechies [15] and Mallat [40] helped complete the framework for the theory and demonstrated the applicability of the theory to mathematical analysis and electrical engineering applications.

Daubechies [15] develops much of the mathematics behind wavelet theory and several families of wavelets are attributed to [15]. In this reference, Daubechies reviews the mathematical basis of multiresolution analysis using both the Laplacian pyramid and the wavelet pyramid. Daubechies [16] makes detailed studies of the similarities and differences between the short–time Fourier transform (STFT) and the WT, and links the theory to time–frequency localization and signal analysis. Nonorthogonal representations are examined and the concept of a frame is developed and discussed. A frame sets an upper bound on the error that results when a signal is reconstructed from its WT coefficients. The decomposition of a function into some wavelet bases is shown to result in tight frames even though the wavelets are not orthogonal.

Burt and Adelson [8] describe a technique for image encoding in which local operators of many scales but identical shape serve as the basis functions. The representations are localized in spatial frequency as well as in space. The representation is referred to as the Laplacian pyramid as the encoding is equivalent to sampling the image with Laplacian operators of many scales. The encoding process enhances salient image features and can be used as the basis for image analysis and compression. Efficient algorithms for coding and decoding are given.

Grossman and Morlet [28] were among the first researchers to investigate the decomposition of arbitrary square integrable real–valued functions into families of square integrable wavelets formed by shifts and dilations from one wavelet. They obtain explicit expressions for the case of a particular analyzing family and make analogies to the coherent states model found in L_2–theory. Grossman [27] points out the applicability of the WT to edge detection, and discusses the multiscale nature of the wavelet transform. He studies an example based on a 1–D signal and demonstrates convergence of different scales of the transform of the signal.

Dutilleux [20] and Holschneider *et al.* [32] present a discrete implementation of the WT originally used for the analysis of sound patterns. This implementation came to be known as the *à trous* implementation, referring to the fact that it contains zeros or *holes* in the filters applied to the signal. Shensa [50] links this implementation to the one given by Mallat and shows their convergence under certain conditions.

Mallat [38, 40, 41] applies the wavelet representation to intensity images and presents an algorithm for taking the 2–D WT and its inverse. Mallat's algorithm decomposes an image into orthogonal components by applying filtering and decimation at each step, keeping the total number of pixels in the representation constant at any scale.

Meyer [44] authored three mathematical volumes on wavelets and operators. They cover several aspects of the mathematical theory as well as applications to pseudodifferential calculus, complex analysis and Hardy spaces, Cauchy operators, and other elements in elliptical and nonlinear partial differential equations. The first volume, entitled "wavelets and operators" addresses the general theory, and was written with mathematicians and engineers in mind. The second and third volumes cover Calderón–Zygmund operators, which are associated with wavelets, and discuss their applications. They are primarily aimed at mathematicians in operator and analysis theory.

1.3 Technical Notations

The following notations will be used in this thesis:

- $L^2(\mathbf{R})$ denotes the Hilbert space of measurable, square–integrable functions, or in signal processing terms the space of *finite energy* signals $f(t)$ satisfying

$$\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty \quad .$$

- The inner product between two functions is denoted by

$$\langle f, g\rangle = \int_{-\infty}^{\infty} f(t) g^*(t) dt \quad .$$

- The symbol $\dagger$ will mean the reversal of the complex conjugate of a filter (the Hermitian adjoint filter), i.e.: $[\boldsymbol{f}^{\dagger}]_k = \bar{f}_{-k}$.

- The symbol $\triangleq$ indicates a sampling.

- The symbol Λ means decimation by a factor of 2.

CHAPTER 2

Time–Frequency Analysis and the Continuous Wavelet Transform (CWT)

The aim of time–frequency signal analysis is to study the spectral behavior of an analog or digital signal by transformations. The Fourier Transform (FT) has been the dominant tool used for analysis of this type. However, it cannot give an accurate result unless full knowledge of the signal in the time domain is available. Thus, the FT is more suited to analysis of *stationary* signals. If the signal is altered in any way, the entire spectrum is affected. Gabor [24] introduced the Gabor transform to account for this deficiency by adding time dependancy in Fourier analysis while preserving linearity. This was done by a time–localization window function through which the signal is approximately stationary. The function used by Gabor was a Gaussian, the transform of which is also a Gaussian. This led to time localization in the inverse transform also.

The Gabor transform can be generalized by using other functions for the localization window. The transform is then referred to in the literature as the short–time Fourier transform (STFT). However, it turns out that it is not possible to find a window with size smaller than or equal to that of the Gaussian functions. This is known as the Uncertainty Principle and imposes limitations on the use of the STFT in the presence of both high and low frequencies.

This led to the introduction of the WT which windows the signal directly, as opposed to the STFT which windows the Fourier transform. This allows the WT to zoom in and out on different parts of the signal. In the remainder of this chapter, we discuss each of these transforms and identify the strengths of the WT for signal analysis and processing.

2.1 The Fourier Transform (FT)

This transform has been used extensively for time–frequency analysis of stationary signals. Given a continuous signal $f(t) \in L^2(\mathbb{R})$, we define the FT as

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \ .$$

As was pointed out earlier, this transform is not suited to the analysis of *nonstationary* signals as any change in time in the nonstationary signal can affect all of the frequency domain representations of the signal. Further, for the purposes of image processing, there is no spatial correspondence between the image and its FT representation. Thus it is impossible to segment an image or locate high frequency features in it by using the FT alone.

2.2 The Gabor Transform, the Short–Time Fourier Transform (STFT), and the Uncertainty Principle

The Gabor transform attempts to establish a *local* representation of a given signal by using a local window and assuming that the function is *stationary* in that window. Given a continuous function or signal, $f(t) \in L^2(\mathbf{R})$, we define the Gabor transform for $\alpha \geq 0$ as

$$(G_b^\alpha)(\omega) = \int_{-\infty}^{\infty} (e^{-j\omega t} f(t)) g_\alpha(t-b) dt \ .$$

This localizes the FT of f around $t = b$. The width of the window is determined by the (fixed) positive constant α.

The above Gabor transform can be generalized by using other functions for the window. This yields the short–time Fourier transform (STFT). The main shortcoming of STFT representations is that resolution in time and frequency cannot be arbitrarily small, because their product is lower bounded [24, 49],

$$\sigma_f \sigma_{\hat{f}} \geq \frac{1}{2} \tag{2.1}$$

where $f \in L^2(\mathbf{R})$, $\hat{f}$ is its Fourier transform, and σ is the width of the function as given by its root mean square deviation. Equality is only attained if the function f is a Gaussian. This is known as the uncertainty principle or Heisenberg inequality. Once a window has been chosen to be used in the STFT, the time–frequency resolution given by Eq. (2.1) is fixed over the entire time–frequency plane (since the same window is used at all frequencies). Thus we can perform analysis with good time resolution or frequency resolution, but not both [49].

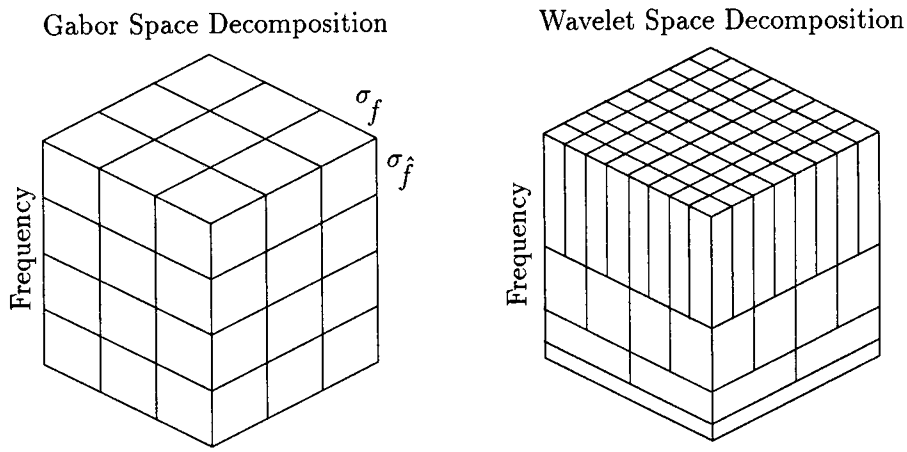


Figure 2.1: *Comparison of Gabor and Wavelet transform time–frequency resolution cells.*

2.3 The Continuous Wavelet Transform (CWT)

The WT overcomes the resolution limitation of the STFT by allowing for different time–frequency localizations. This is done by using basis functions that are formed by *dilations* as well as *translations* of a prototype function $\psi(t)$. These dilations can generate both high–frequency, short–duration and low–frequency, long–duration functions that are better suited for representing signals than the Gabor representation. Figure 2.1 shows the time–frequency resolution cells for both STFT and the WT.

The uncertainty principle still holds, but we have different combinations for σ_f and $\sigma_{\hat{f}}$. Depending on the wavelets used, we can get very close, but not equal to the lowest limit given by the Gaussian function.

This idea of taking a signal and analyzing it at various scales and resolutions has existed for a long time in mathematics and physics. In fact, similar ideas were proposed as early as in 1910 [29], and other literature appeared later in [9, 22, 37, 49]. In the section below, the CWT is defined and compared with the previously discussed transforms.

2.3.1 Definition

Given a 1–D signal *f(t)*$\in L^2(\mathbf{R})$, the WT is given by the inner product

$$(W_\psi f)(b,a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t)\psi^*\left(\frac{t-b}{a}\right) dt \quad a, b \in \mathbf{R}, a \neq 0 \ . \tag{2.2}$$

The function ψ is called the *basic* or *mother* wavelet. The minimal assumptions imposed on ψ are [27]

i) $$\int_{-\infty}^{\infty} |\psi(x)|^2 dx < \infty \tag{2.3}$$

ii) $$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \tag{2.4}$$

The square integrability condition on ψ ensures that the local average value decays to zero at $\pm\infty$. The wavelets actually used in this work satisfy much stronger decay conditions. By imposing this first condition, the second condition becomes equivalent to $\hat{\psi}(0) = 0$, or $\int_{-\infty}^{\infty} \psi(x)\ dx = 0$. This condition is necessary for the transform to have an inverse and is referred to as the wavelet admissibility condition [17]. In practice, we impose much stronger conditions on the decay of the mother wavelet ψ, especially in discrete cases as with the wavelets used in this work.

If we set

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

we can write the CWT as the inner product

$$(W_{\psi}f)(b,a) = \langle f, \psi_{a;b}\rangle .$$

There are several ways of looking at the transform. One way is to view a wavelet as a band–pass function. In this case, the CWT appears as a filtered version of the signal. Another view is to think of wavelets as basis functions. The CWT then appears as a decomposition onto the wavelet space. These basis functions are scaled and translated versions of the mother wavelet. These different views are useful in the analysis of the properties of the CWT, and lead to different applications for the WT.

2.3.2 The CWT in Higher Dimensions

There are different methods of extending the CWT to higher dimensions. The first requirement is obtaining multidimensional extensions to wavelets. One way of doing this is using inner products of single dimensional wavelets and scaling functions. These are called separable wavelets. Mallat's multiresolution decomposition and the *à trous* algorithm are examples of this approach. These will be discussed in Chapter 3 for the discrete case. A different approach is to use true (nonseparable) multidimensional wavelets and scaling functions [34]. This approach was not used for our purposes and will not be discussed.

2.4 Multiresolution Analysis

Another way of introducing wavelets and the CWT is through multiresolution analysis. In this section, the concept of multiresolution analysis is introduced, then linked to the CWT.

2.4.1 The Closed Subspaces V_j and Scaling Functions

A multiresolution analysis of $L^2(\mathbb{R})$ is defined as a sequence of closed subspaces V_j of $L^2(\mathbb{R})$, j $\in \mathbb{Z}$, with the following properties [17]:

1. $V_{j+1} \subset V_j \ ,$ (2.5)

2. $\overline{\bigcup_{j=-\infty}^{+\infty} V_j} = L^2(\mathbb{R}) \ ,$ (2.6)

3. $\bigcap_{j=-\infty}^{+\infty} V_j = \{0\} \ ,$ (2.7)

4. $f \in V_j \Longleftrightarrow f(2^j.) \in V_0 \ .$ (2.8)

Thus all the spaces are scaled versions of the central space V_0. The last condition implies that if $f \in V_j$, then $f(. - 2^j n) \ \in V_j, \ \ \forall n \in \mathbb{Z}$. We also define a projection operator P_j such that $P_j f$ maps f onto V_j. By virtue of Eq. (2.6) above, we also have $\lim_{j \to -\infty} P_j f = f$. The simplest example of spaces V_j satisfying the above criteria is the Haar system. It is given by

$$V_j = \{f \in L^2(\mathbb{R}); \quad \forall k \in \mathbb{Z} : f|_{[2^j k, 2^j (k+1)[} = constant\} \ .$$

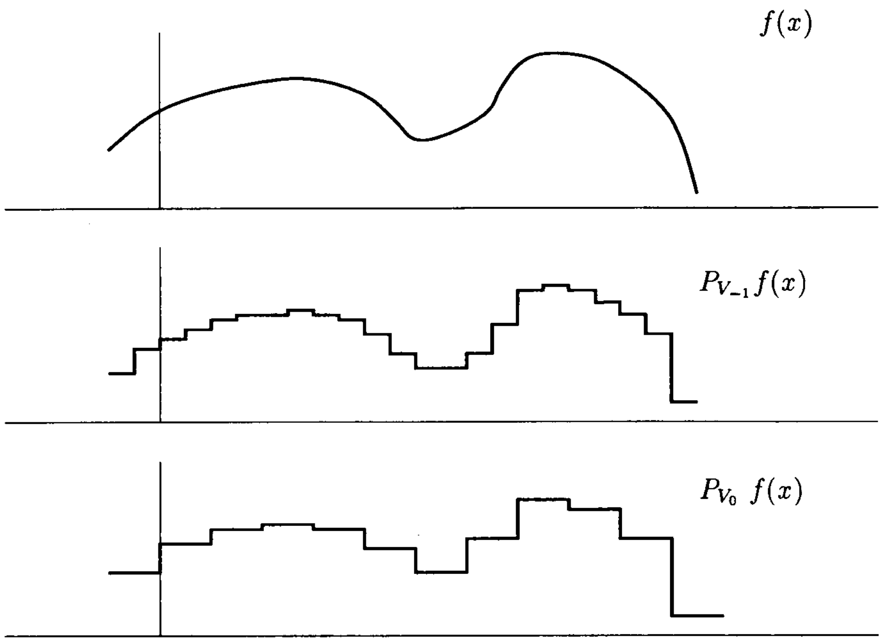


Figure 2.2: *Projection of a function $f(x)$ onto the Haar spaces V_{-1} and V_0.*

Figure 2.2 shows the projection of a function $f(x)$ on the Haar spaces.

We now introduce a function ϕ to be called the scaling function of the multiresolution analysis such that $\phi \in V_0$ and

$$\{\phi_{0,n}; n \in \mathbf{Z}\} \text{ is an orthonormal basis in } V_0 \ ,$$

where $\phi_{j,n}(x) = 2^{-j/2}\phi(2^{-j}x - n)$.

Given these conditions, we can summarize the properties of a multiresolution analysis as follows.

1. **Two–scale Difference Equation**

Since $\phi \in V_0 \subset V_1$, and $\phi_{-1,n}$ are an orthonormal basis in V_{-1}, we have

$$\phi = \sum_n h_n \phi_{-1,n} \ .$$

Thus the scaling function can be written as a *dilation* or a *two–scale difference equation*

$$\phi(x) = \sqrt{2} \sum_n h_n \phi(2x - n) \tag{2.9}$$

with

$$h_n = \langle \phi, \phi_{-1,n} \rangle \text{ and } \sum_{n \in \mathbb{Z}} |h_n|^2 = 1.$$

2. **Calculation of Scaling Function**

The normalization

$$\int_{-\infty}^{\infty} \phi(x) dx = 1 \ ,$$

along with the dilation equation above uniquely identify a scaling function. Usually, there is no explicit expression for ϕ but there are algorithms for calculating the scaling function ϕ at all the dyadic points ($x = 2^{-j}k; j, k \in \mathbb{Z}$) from the refinement equation. In practice, it is not necessary to calculate the scaling function itself as we can work directly with the basis coefficients h_n.

3. **Fourier Transform of Scaling Function**

The FT of the scaling function satisfies

$$\hat{\phi}(\omega) = m_0(\xi/2)\hat{\phi}(\xi/2)$$

where

$$m_0(\xi) = \frac{1}{\sqrt{2}} \sum_n h_n e^{-in\xi}$$

m_0 is a 2π−periodic function in $L^2([0, 2\pi])$.

4. **Translation of Scaling Function**

The scaling function ϕ and its integer translates form a *partition of unity*:

$$\forall x \in \mathbb{R} : \sum_n \phi(x - n) = 1 \ .$$

This follows from

$$\begin{cases} \hat{\phi}(0) = 1 \\ \hat{\phi}(2\pi k) = 0; k \in \mathbb{Z}, k \neq 0 \end{cases} \tag{2.10}$$

and Poisson's summation formula

$$\sum_{k=-\infty}^{\infty} f(x + 2\pi k) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \hat{f}(k) e^{ikx}, \quad x \in \mathbb{R} \ .$$

In particular,

$$\sum_{k=-\infty}^{\infty} f(2\pi k) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \hat{f}(k) \ .$$

2.4.2 The Detail Spaces W_j and Wavelet Functions

Let us now define the spaces W_j such that for every $j \in \mathbb{Z}$, W_j is the orthogonal complement of V_j in V_{j-1}, we have

$$V_{j-1} = V_j \oplus W_j \tag{2.11}$$

and

$$W_j \perp W_{j'} \quad \forall j' \neq j \ . \tag{2.12}$$

Keeping in mind Eqs. (2.6) and (2.7) above, we get

$$L^2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_j \tag{2.13}$$

which is a decomposition of $L^2(\mathbb{R})$ into mutually orthogonal subspaces [17]. Also, the W_j inherit the scaling property given in Eq. (2.8) from the V_j:

$$f \in W_j \iff f(2^j.) \in W_0 \ .$$

A function ψ is a wavelet if, for fixed j, $\{\psi_{j,k}; k \in \mathbb{Z}\}$ constitutes an orthonormal basis for W_j. By virtue of Eqs. (2.13) and (2.6), the whole collection $\{\psi_{j,k}; j, k \in \mathbb{Z}\}$ is an orthonormal basis for $L^2(\mathbb{R})$. We note the following with regard to this function.

1. $\psi \in W_0$, which is equivalent to $\psi \in V_{-1}$ and $\psi \perp V_0$. Since it is in V_{-1}, we can write it as

$$\psi(x) = \sqrt{2}\sum_n g_n \phi(2x - n) \ .$$

2.The FT of ψ is given by

$$\hat{\psi}(\xi) = \nu(\xi/2)\hat{\phi}(\xi/2) \ ,$$

where

$$\nu(\xi) = \frac{1}{\sqrt{2}}\sum_n g_n e^{-in\xi} \ ,$$

and ν is again $2\pi-$periodic function.

3. The wavelet ψ satisfies

$$\int_{-\infty}^{\infty} \psi(x)dx = 0 \ .$$

The coefficients g_n satisfy

$$\sum_n g_n = 0 \ .$$

The following theorem, from [17], sums up how wavelets fit in multiresolution analysis.

Theorem 2.1: If a ladder of closed subspaces, $(V_j)_{j\in\mathbb{Z}}$ in $L^2(\mathbb{Z})$, satisfies the conditions 2.5–2.8, then there exists an associated orthonormal wavelet basis $\{\psi_{j,k}; j,k \in \mathbb{Z}\}$ for $L^2(\mathbb{R})$ such that

$$P_{j-1} = P_j + \sum_n \langle \cdot, \psi_{j,k}\rangle \psi_{j,k} \ . \tag{2.14}$$

If a wavelet ψ satisfies Eq. (2.14) above, then so does any wavelet ψ_g that satisfies

$$\hat{\psi}_g(\xi) = \rho(\xi)\hat{\psi}(\xi) \ ,$$

where ρ is 2π – periodic and $|\rho(\xi)| = 1$. Thus, in contrast with the scaling function, a wavelet function is not determined uniquely for a given multiresolution analysis ladder. One possible choice is

$$\psi = \sum_n g_n \phi_{-1,n} \text{ with } g_n = (-1)^n h_{-n+1}$$

or

$$g_n = (-1)^n h_{-n+1+2N}$$

with $N \in \mathbb{Z}$ chosen depending on the number of samples we use for h and g.

CHAPTER 3

The Discrete Wavelet Transform (DWT)

In signal processing, we usually work with discrete signals that have been obtained by uniform sampling of a continuous signal. Thus, a discrete version of the WT is needed. There have been two main implementations for the DWT. One of these is the *algorithme à trous* which was developed for music synthesis [20] and used with nonorthogonal wavelets. The other is Mallat's multiresolution approach [38, 40] which was originally used in two dimensions for image processing and employed orthonormal wavelets. In this chapter, we discuss and compare these algorithms and link them to the CWT. We also highlight the features of each implementation that are useful for computer vision purposes.

3.1 Mallat's Multiresolution Algorithm

3.1.1 A Multiresolution Transform

The concept of multiresolution analysis was already introduced in Chapter 2 in the context of the CWT. In this section, we discuss the discrete implementation of this concept and show examples of its application to functions. We will restrict ourselves to *dyadic* scales with a dilation factor of 2. The notation (V_{2^j}) is used to denote the closed subspaces of a multiresolution ladder.

Let (V_{2^j}) be a multiresolution approximation of $L^2(\mathbb{R})$ and let $\phi(x) \in L^2(\mathbb{R})$ be the associated scaling function. If we set $\phi_{2^j} = 2^j\phi(2^jx)$ for $j \in \mathbb{Z}$, then

$$(\sqrt{2^{-j}}\phi_{2^j}(x - 2^{-j}n))_{n\in\mathbf{Z}} \text{ is an orthonormal basis of } V_{2^j} \tag{3.1}$$

as was indicated in Chapter 2. Thus we can build an orthonormal basis of any V_{2^j} by dilating a function $\phi(x)$ with a coefficient 2^j and translating the resulting function on a grid whose interval is proportional to 2^{-j} [40]. For any given multiresolution approximation $(V_{2^j})_{n\in\mathbf{Z}}$, there exists a unique scaling function $\phi(x)$ which satisfies Eq. (3.1) above.

We can compute the orthogonal projection on V_{2^j} by decomposing a signal $f(x)$ on the orthonormal basis given in Eq. (3.1). Specifically,

$$\forall f(x) \in L^2(\mathbb{R}), \quad A_{2^j}f(x) = 2^{-j} \sum_{n=-\infty}^{+\infty} \langle f(u), \phi_{2^j}(u - 2^{-j}n)\rangle \phi_{2^j}(x - 2^{-j}n)\ . \tag{3.2}$$

$A_{2^j}f(x)$ is the approximation of the signal $f(x)$ at the resolution 2^j. It is characterized by the set of inner products

$$A^d_{2^j}f = (\langle f(u), \phi_{2^j}(u - 2^{-j}n)\rangle)_{n\in\mathbf{Z}}\ .$$

$A^d_{2^j}f$ is called a *discrete* approximation of $f(x)$ at the resolution 2^j. Now each of the inner products can be interpreted as a convolution product evaluated at a point $2^{-j}n$

$$\begin{aligned} &\langle f(u), \phi_{2^j}(u - 2^{-j}n)\rangle \\ &= \int_{-\infty}^{+\infty} f(u)\phi_{2^j}(u - 2^{-j}n)du \\ &= (f(u) * \phi_{2^j}(-u))(2^{-j}n) \end{aligned}$$

Hence we can rewrite $A^d_{2^j} f$

$$A^d_{2^j} f = ((f(u) * \phi_{2^j}(-u))(2^{-j}n))_{n \in \mathbf{Z}} \ .$$

Since the scaling function $\phi(x)$ is a low–pass filter, this discrete signal can be interpreted as a low–pass filtering of the signal $f(x)$ followed by a uniform sampling at the rate 2^j.

3.1.2 Implementation of a Multiresolution Transform

In general, we can measure a signal only as a discrete function. Thus our starting point is itself a discrete approximation. If we let this measured signal be $A^d_1 f$, then we would like to calculate all the discrete approximation $A^d_{2^j}$ for $j < 0$. The following method uses an iterative technique [40].

Let $(V_{2^j})_{j \in \mathbf{Z}}$ be a multiresolution approximation and $\phi(x)$ be the corresponding scaling function. From $\phi_{2^j}(x - 2^{-j}n) \in V_{2^j}$ and $V_{2^j} \subset V_{2^{j+1}}$, we can expand this function in $V_{2^{j+1}}$ as follows:

$$\phi_{2^j}(x - 2^{-j}n) = 2^{-j-1} \sum_{k=-\infty}^{+\infty} \langle \phi_{2^j}(u - 2^{-j}n), \phi_{2^{j+1}}(u - 2^{-j-1}k) \rangle . \phi_{2^{j+1}}(x - 2^{-j-1}k) \ . \tag{3.3}$$

A change of variables leads to

$$2^{-j-1} \langle \phi_{2^j}(u - 2^{-j}), \phi_{2^{j+1}}(u - 2^{-j-1}k) \rangle = \langle \phi_{2^{-1}}(u), \phi(u - (k - 2n)) \rangle \ .$$

When computing the inner products of both sides of Eq. (3.3) we obtain

$$\langle f(u), \phi_{2^j}(u - 2^{-j}) \rangle = \sum_{k=-\infty}^{+\infty} \langle \phi_{2^{-1}}(u), \phi(u - (k - 2n)) \rangle . \langle f(u), \phi_{2^{j+1}}(u - 2^{-j-1}k) \rangle \ . \tag{3.4}$$

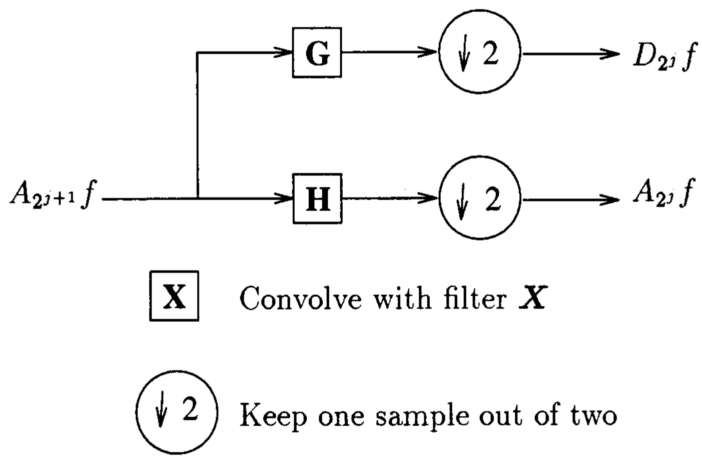


Figure 3.1: *A single stage in Mallat's multiresolution algorithm [40].*

Let H be the discrete filter whose impulse response is given by

$$\forall n \in \mathbb{Z}, h(n) = \langle \phi_{2^{-1}}(u), \phi(u-n) \rangle \ . \tag{3.5}$$

Let $\tilde{H}$ be the mirror filter with impulse response $\tilde{h}(n) = h(-n)$. By inserting Eq. (3.5) in the previous equation (Eq. (3.4)), we obtain

$$\langle f(u), \phi_{2^j}(u - 2^{-j}n) \rangle = \sum_{k=-\infty}^{+\infty} \tilde{h}(2n-k) \langle f(u), \phi_{2^{j+1}}(u - 2^{-j-1}k) \rangle \ . \tag{3.6}$$

Equation (3.6) shows that $A^d_{2^j}$ can be computed by convolving $A^d_{2^{j+1}}$ with $\tilde{H}$ and keeping every other sample of the output. This can be extended for all $A^d_{2^j}$ for $j < 0$. A single stage of this algorithm is shown in Fig. 3.1.

Mallat [40] gives a practical characterization of the FT of a scaling function as follows.

Let $\phi(x)$ be a scaling function, and let H be a discrete filter with impulse response

$h(n) = \langle \phi_{2^{-1}}(u), \phi(u-n) \rangle$. Let $H(\omega)$ be the Fourier series defined by

$$H(\omega) = \sum_{n=-\infty}^{+\infty} h(n) e^{-in\omega} .$$

$H(\omega)$ satisfies the following two properties:

- $|H(0)| = 1$ and $h(n) = O(n^{-2})$ at infinity , (3.7)

- $|H(\omega)|^2 + |H(\omega + \pi)|^2 = 1$. (3.8)

Conversely, let $H(\omega)$ be a Fourier series satisfying Eqs. (3.7) and (3.8), such that

$$|H(\omega)| \neq 0 \text{ for } \omega \in \left[0, \frac{\pi}{2}\right] . \tag{3.9}$$

Then the function defined by

$$\hat{\phi}(\omega) = \prod_{p=1}^{+\infty} H(2^{-p}\omega) \tag{3.10}$$

is the FT of a scaling function. Equation (3.10) allows us to construct a scaling function $\phi(x)$ starting from a filter function $H(\omega)$ that satisfies Eqs. (3.7)–(3.9). It is possible to choose $H(\omega)$ in order to obtain a scaling function $\phi(x)$ which has good localization properties in both the frequency and spatial domains [40]. The smoothness class of $\phi(x)$ and its asymptotic decay at infinity can be estimated from the properties of $H(\omega)$ [15]. The scaling function gives us the signal at successive resolutions. In the next section, a multiresolution representation based on the *difference* of information available at successive resolutions is computed.

3.1.3 The Orthogonal Wavelet Representation

The difference of information between the approximation of a function $f(x)$ at the resolutions 2^{j+1} and 2^j is called the *detail signal* at the resolution 2^j. The approximations at these resolutions is given by the orthogonal projections on the subspaces $V_{2^{j+1}}$ and V_{2^j} respectively. It can be shown [40] that the detail signal at the resolution 2^j is given by the orthogonal projection of the original signal on the orthogonal complement of V_{2^j} in $V_{2^{j+1}}$. If we let O_{2^j} be this orthogonal complement, i.e.:

$$O_{2^j} \text{ is orthogonal to } V_{2^j}$$

$$O_{2^j} \oplus V_{2^j} = V_{2^{j+1}} \ .$$

To compute the orthogonal projection of a function $f(x)$ on O_{2^j}, we need an orthonormal basis of O_{2^j}. It turns out that such a basis can can be built by scaling and translating a function $\psi(x)$. Mallat [40] gives the following formula for constructing $\psi(x)$:

$$\hat{\psi}(\omega) = G\left(\frac{\omega}{2}\right)\hat{\phi}\left(\frac{\omega}{2}\right) \text{ with } G(\omega) = e^{-i\omega}\overline{H(\omega+\pi)} \ . \tag{3.11}$$

If we let $\psi_{2^j}(x) = 2^j\psi(2^j x)$ denote dilation, then

$$(\sqrt{2^{-j}}\psi_{2^j}(x - 2^{-j}n))_{n\in\mathbf{Z}} \text{ is an orthonormal basis of } O_{2^j}$$

and

$$(\sqrt{2^{-j}}\psi_{2^j}(x - 2^{-j}n))_{(n,j)\in\mathbf{Z}} \text{ is an orthonormal basis of } L^2(\mathbf{R}) \ .$$

The proof for this fundamental result is given in [40]. It allows us to construct a wavelet $\psi(x)$ from $H(\omega)$ and $\phi(x)$. Daubechies [15] related the properties of the scaling function and the wavelet to the filter function H and showed that for any $n > 0$, we can find a function $H(\omega)$ such that the corresponding wavelet $\psi(x)$ has compact support and is n times continuously differentiable.

3.1.4 Implementation of an Orthogonal Wavelet Representation

If we follow the same derivation steps as in Section 3.1.2 but for ψ, we can show that $D_{2^j} f$ can be computed by convolving $A^d_{2^{j+1}} f$ with a discrete filter G as follows: Let G be the discrete filter with impulse response

$$g(n) = \langle \psi_{2^{-1}}(u), \phi(u-n) \rangle \ , \tag{3.12}$$

and $\tilde{G}$ be the symmetric filter with impulse response $\tilde{g}(n) = g(-n)$. By analogy with Eq. (3.6), we get

$$\langle f(u), \phi_{2^j}(u - 2^{-j}n) \rangle = \sum_{k=-\infty}^{+\infty} \tilde{h}(2n-k) \langle f(u), \phi_{2^{j+1}}(u - 2^{-j-1}k) \rangle \ . \tag{3.13}$$

Again, Eq. (3.13) tells us that we can compute the detail signal $D_{2^j} f$ by convolving $A^d_{2^{j+1}} f$ with the filter $\tilde{G}$ and retaining every other sample of the output. Theorem 1 (Section 2.4.1) tells us that the impulse response of filter G is related to the impulse response of filter H by

$$g(n) = (-1)^{1-n} h(1-n) \ . \tag{3.14}$$

G is the *mirror* filter of H and is a high–pass filter. The filters G and H are called *quadrature mirror* filters. If the original signal has N samples, then the discrete signals $D_{2^j} f$ and $A^d_{2^j} f$ have $2^j N$ samples each. Thus the wavelet representation

$$(A^d_{2^{-J}} f, (D_{2^j} f)_{-J \leq j \leq -1})$$

has the same total number of samples as the original approximated signal $A^d_1 f$. This is due to the representation being orthogonal.

3.1.5 Extension of Wavelet Representations to Images

The wavelet representation can be extended for higher dimensions. In this section we present one extension to image analysis. A multiresolution approximation of $L^2(\mathbb{R}^2)$ is a sequence of subspaces of $L^2(\mathbb{R}^2)$ which satisfies a straightforward 2–D extension of the properties in Eqs. (2.5)–(2.8) [40]. Let $(V_{2^j})_{j\in\mathbf{Z}}$ be a multiresolution approximation of $L^2(\mathbb{R}^2)$ satisfying those properties. As in the 1–D case, it can be shown that a unique scaling function $\Phi(x,y)$ exists such that its dilations and translations give an orthonormal basis for each space (V_{2^j}). We denote the dilation of a 2–D function by

$$\Phi_{2^j}(x,y) = 2^{2j}\Phi(2^j x, 2^j y).$$

The family of functions

$$(2^{-j}\Phi_{2^j}(x - 2^{-j}n, y - 2^{-j}m))_{(n,m)\in\mathbf{Z}^2}$$

form an orthonormal basis of (V_{2^j}). Mallat's algorithm [40] applies to the particular case of separable multiresolution approximations of $L^2(\mathbb{R}^2)$. Here, each vector subspace $V_{2^j} \in L^2(\mathbb{R}^2)$ can be decomposed as a tensor product of two identical subspaces of $L^2(\mathbb{R})$ [40]:

$$V_{2^j} = V_{2^j}^1 \otimes V_{2^j}^1 \ .$$

The sequence $(V_{2^j})_{j\in\mathbf{Z}}$ forms a multiresolution approximation if and only if the sequence $(V_{2^j}^1)_{j\in\mathbf{Z}}$ forms a multiresolution approximation of $L^2(\mathbb{R})$. The scaling function can be written as

$$\Phi(x,y) = \phi(x)\phi(y) \ ,$$

where ϕ is the 1–D scaling function of the multiresolution approximation $(V^1_{2^j})_{j\in\mathbf{Z}}$.

The following theorem extends Theorem 2.1 to 2–D.

Theorem 3.1 [40]: Let $(V_{2^j})_{j\in\mathbf{Z}}$ be a separable multiresolution approximation of $L^2(\mathbb{R}^2)$. Let $\Phi(x,y) = \phi(x)\phi(y)$ be the associated 2–D scaling function. Let $\psi(x)$ be the 1–D wavelet associated with the scaling function $\phi(x)$. Then the three wavelets

$$\Psi^1(x,y) = \phi(x)\psi(y)$$

$$\Psi^2(x,y) = \psi(x)\phi(y)$$

$$\Psi^3(x,y) = \psi(x)\psi(y)$$

are such that

$$\begin{aligned}(\quad & 2^{-j}\,\Psi^1_{2^j}(x-2^{-j}n, y-2^{-j}m),\\ & 2^{-j}\,\Psi^2_{2^j}(x-2^{-j}n, y-2^{-j}m),\\ & 2^{-j}\,\Psi^3_{2^j}(x-2^{-j}n, y-2^{-j}m) \quad)_{(n,m)\in\mathbf{Z}^2}\end{aligned}$$

is an orthonormal basis of O_{2^j}, and since $L^2(\mathbb{R}^2)$ can be decomposed as a direct sum of the orthogonal spaces O_{2^j} ,

$$L^2(\mathbb{R}^2) = \bigoplus_{j=-\infty}^{+\infty} O_{2^j}$$

the family of functions

$$\begin{aligned}(\quad & 2^{-j}\,\Psi^1_{2^j}(x-2^{-j}n, y-2^{-j}m),\\ & 2^{-j}\,\Psi^2_{2^j}(x-2^{-j}n, y-2^{-j}m),\\ & 2^{-j}\,\Psi^3_{2^j}(x-2^{-j}n, y-2^{-j}m) \quad)_{j\in\mathbf{Z},(n,m)\in\mathbf{Z}^2}\end{aligned}$$

is an orthonormal basis of $L^2(\mathbb{R}^2)$.

This theorem tells us that the difference of information between $A^d_{2^{j+1}}$ and $A^d_{2^j}$ is given by three detail images $D^1_{2^j}f$, $D^2_{2^j}f$, and $D^3_{2^j}f$. Just as in 1–D signals, the inner products which define $A^d_{2^j}$, $D^1_{2^j}f$, $D^2_{2^j}f$, and $D^3_{2^j}f$ are equal to a uniform sampling of 2–D convolution–products

$$A^d_{2^j}f = ((f(x,y) * \phi_{2^j}(-x)\phi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)\in\mathbf{Z}^2} \,, \tag{3.15}$$

$$D^1_{2^j}f = ((f(x,y) * \phi_{2^j}(-x)\psi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)\in\mathbf{Z}^2} \,, \tag{3.16}$$

$$D^2_{2^j}f = ((f(x,y) * \psi_{2^j}(-x)\phi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)\in\mathbf{Z}^2} \,, \tag{3.17}$$

$$D^3_{2^j}f = ((f(x,y) * \psi_{2^j}(-x)\psi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)\in\mathbf{Z}^2} \,. \tag{3.18}$$

All of the above expressions can be calculated with separable filtering of the signal along the rows and columns as shown in Fig. 3.2.

Figure 3.3 shows the application of Mallat's algorithm for a 2–D image using Daubechies wavelets of order 4.

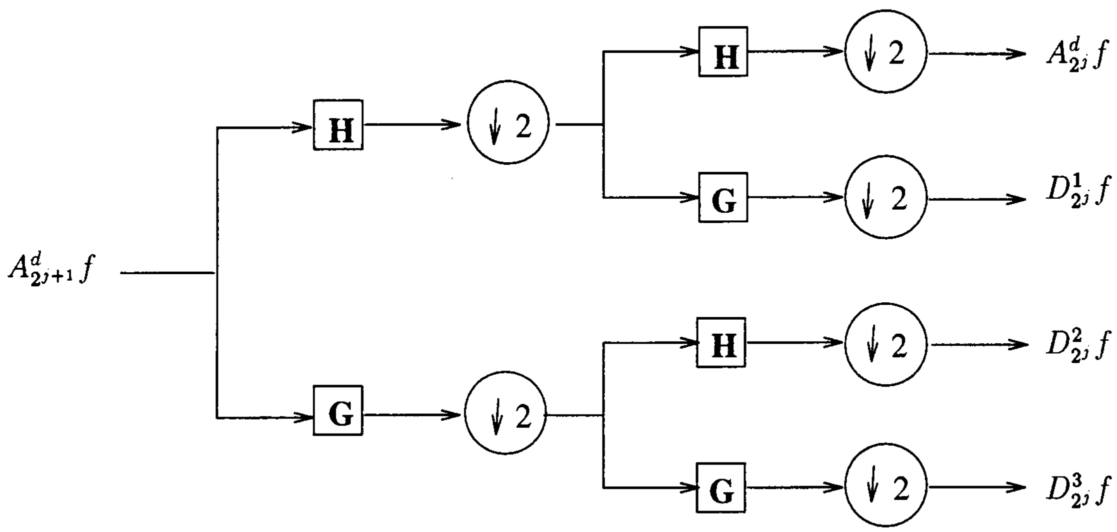


Figure 3.2: *A stage in Mallat's multiresolution algorithm for images.*

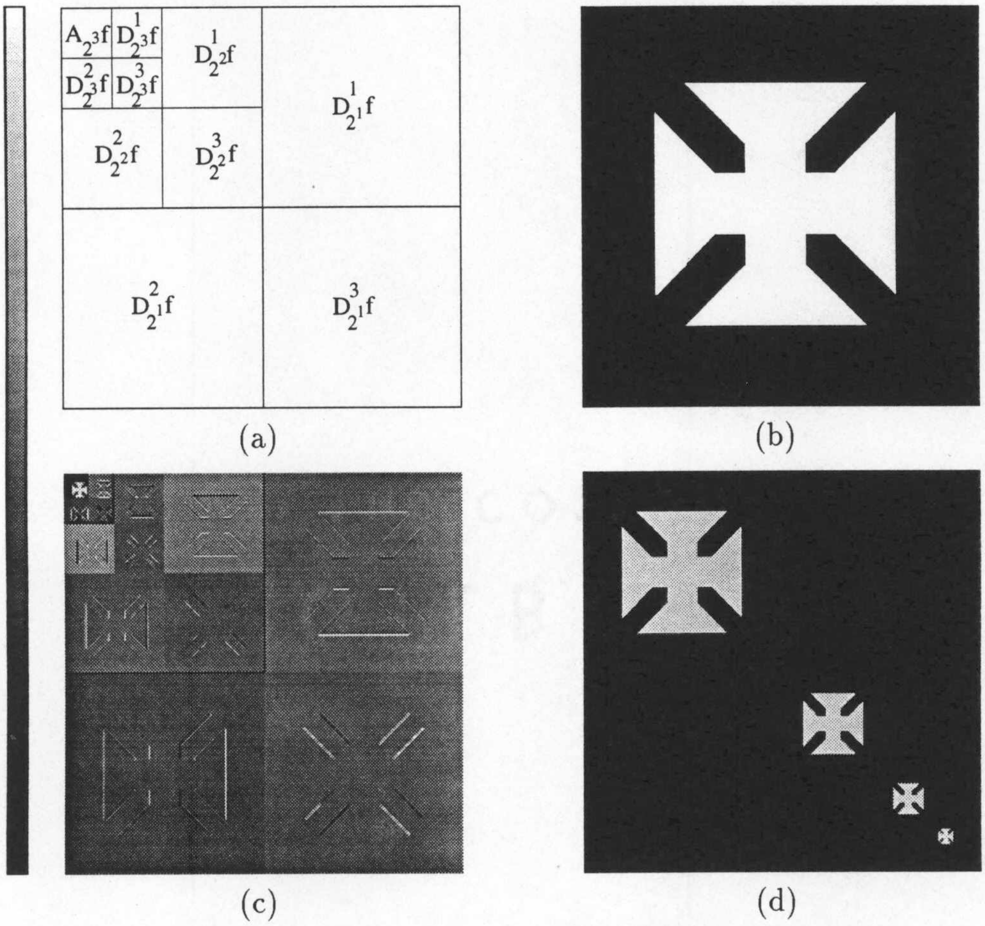


Figure 3.3: *The 2–D DWT: (a) Schematic representation of the 2–D DWT, (b) Original image, (c) Wavelet decomposition into three multiresolution layers, and (d) Approximation of image at three resolutions.*

3.2 The À *Trous* Algorithm

The *à trous* algorithm [20] was originally devised to be an efficient implementation of the DWT. It can be viewed as a nonorthonormal multiresolution algorithm for which the DWT is exact [50]. This algorithm exists in two forms, decimated and undecimated. The decimated algorithm is identical to Mallat's algorithm [40] except for the constraints on their filters. However, the *à trous* algorithm was developed for nonorthogonal wavelets for use in signal processing, while the multiresolution approach of Mallat was based on orthonormal wavelets. In the following section, we start with the decimated *à trous* algorithm and derive a formal definition for an undecimated DWT.

3.2.1 The Decimated À *Trous* Algorithm

We start with the CWT of a signal $s(t)$

$$W(a,b) = a^{-\frac{1}{2}} \int \bar{\psi}\left(\frac{t-b}{a}\right) s(t)\, dt \ , \tag{3.19}$$

where ψ is the analyzing wavelet, a is the dilation factor, and b is a translation in time. This transform is discretized by assuming that $a = 2^i$ and taking b to be a multiple of a. The choice for the translation factor b is critical, as undersampling leads to incompleteness while oversampling results in a redundant set of functions [50]. The choice $b = na$ is taken from the orthonormal algorithm as it ensures the invertibility of the transform [40]. The WT takes the form

$$W(2^i, 2^i n) \triangleq 2^{-\frac{1}{2}i} \int \bar{\psi}\left(\frac{t}{2^i} - n\right) s(t)\, dt \ . \tag{3.20}$$

Finally, we discretize the integral into a sum:

$$W(2^i, 2^i n) \triangleq 2^{-\frac{1}{2}i} \sum_k \bar{\psi}\left(\frac{k}{2^i} - n\right) s(t)\ dt\ . \tag{3.21}$$

The difficulty in implementing Eq. (3.21) is that as i increases, $\bar{\psi}(t)$ must be sampled at more points, which can be a computational burden. This was overcome in [32] by using an interpolater. This interpolater approximates the values at the nonintegral parts using a finite filter. The interpolation can be demonstrated with the following example [50]. Let $\psi(t)$ be a wavelet known at $t = n$ for $n \in \mathbb{Z}$. Let $\boldsymbol{f}^\dagger$ be the interpolating filter $(0.5, 1.0, 0.5)$. Then

$$\sum_k f_{n-2k}\psi(k) = \begin{cases} \psi\left(\frac{n}{2}\right)\ , & n \text{ even} \\ \frac{1}{2}\left(\psi\left(\frac{n-1}{2}\right) + \psi\left(\frac{n+1}{2}\right)\right)\ , & n \text{ odd} \end{cases} \tag{3.22}$$

approximates a sampling of $\psi(t/2)$. The general procedure is shown in Fig. 3.4 where $\boldsymbol{D}$ is a dilation operator. This operator inserts zeros in the original filter to make space for the interpolated values. The interpolating filter $\boldsymbol{f}^\dagger$ leaves the even points fixed and interpolates at the odd points. The following is a formal definition.

Definition 1: A function is said to be an *à trous* filter if it satisfies

$$f_{2k} = \delta(k)/\sqrt{2}\ . \tag{3.23}$$

The $\sqrt{2}$ is the normalization factor that appeared in Eq. (3.21). Let $\boldsymbol{g}$ be the filter defined by $g_n^\dagger \triangleq \psi(n)$, or equivalently $g_n = \bar{\psi}(-n)$. The operation is given by [50]

$$[\boldsymbol{f}^\dagger * (\boldsymbol{D}\boldsymbol{g}^\dagger)]_n \quad = \quad [\boldsymbol{F}^\dagger \boldsymbol{D}\boldsymbol{g}^\dagger)]_n$$

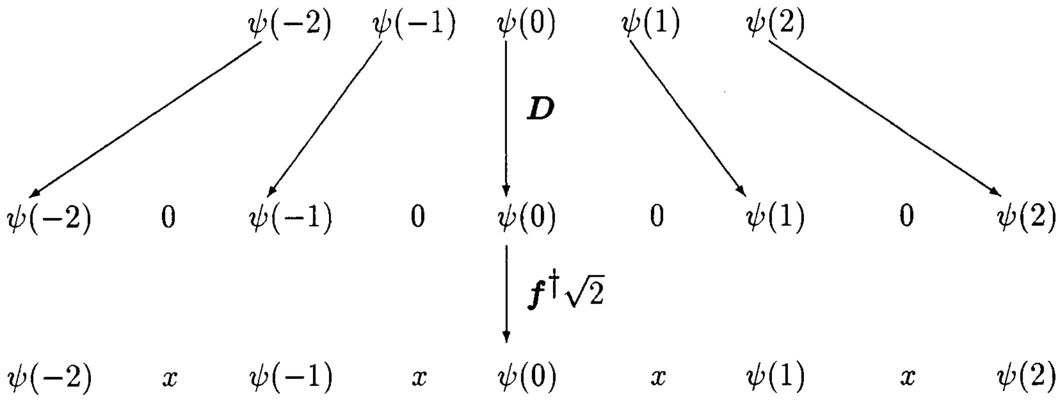


Figure 3.4: *Dilation of a function using an interpolating filter*

$$\begin{aligned} &= \sum_k f^\dagger_{n-2k}\psi(k) \\ &\approx \frac{1}{\sqrt{2}}\psi(n/2)\ . \end{aligned} \tag{3.24}$$

Rewriting $\psi(\frac{k}{2}-n)=\psi\left(\frac{k-2n}{2}\right)$, and inserting Eq. (3.24) into Eq. (3.21) gives

$$\begin{aligned} w(2,2n) &\approx \sum_{k,m} f^\dagger_{k-2n-2m}\bar{g}^\dagger_m s_k \\ &= \sum_{k,m'} g_{n-m'}\bar{f}_{2m'-k}s_k \\ &= [\boldsymbol{g}*(\Lambda(\bar{\boldsymbol{f}}*\boldsymbol{s}))]_n\ , \end{aligned} \tag{3.25}$$

which is w^i_n with $i=1$ as given by Eq. (3.21). Applying mathematical induction to Eq. (3.25) and dropping the n, we get

$$\boldsymbol{s}^{i+1} = \Lambda(\boldsymbol{f}*\boldsymbol{s}^i)\ , \tag{3.26}$$

$$\boldsymbol{w}^i = \boldsymbol{g}*\boldsymbol{s}^i\ . \tag{3.27}$$

This is the decimated *à trous* algorithm. It is a DWT for which the filter $\boldsymbol{f}$ satisfies Eq. (3.23) and the filter $\boldsymbol{g}$ is obtained by sampling an *à priori* wavelet function $\psi(t)$ [50]. In comparison, Mallat's algorithm is

$$\boldsymbol{s}^{i+1} = \Lambda(\boldsymbol{h} * \boldsymbol{s}^i) , \tag{3.28}$$

$$\boldsymbol{d}^{i+1} = \Lambda(\boldsymbol{g} * \boldsymbol{s}^i) , \tag{3.29}$$

where the $\boldsymbol{f}$ has been replaced by $\boldsymbol{h}$, indicating that the latter is in fact not an interpolater but a constrained filter which is related to $\boldsymbol{g}$ as we saw in Section 2.5.3. The following conditions on $\boldsymbol{h}$ and $\boldsymbol{g}$ gaurantee an orthonormal multiresolution analysis.

$$\sum_j [\bar{h}_{2j-n} h_{2j-m} + \bar{g}_{2j-n} g_{2j-m}] = \delta_{nm} , \tag{3.30}$$

$$\sum_j h_{2n-j} \bar{g}_{2m-j} = 0 , \tag{3.31}$$

$$\sum_n g_n = 0 , \tag{3.32}$$

$$\sum_n h_n = \sqrt{2} . \tag{3.33}$$

From Eqs. (3.30) and (3.31) and recalling $H_{mn} \triangleq h_{m-n}$ and $\boldsymbol{\Lambda}^\dagger = \boldsymbol{D}$, we can rewrite Eqs. (3.30) and (3.31) as

$$(\boldsymbol{\Lambda H})(\boldsymbol{H}^\dagger \boldsymbol{D}) = \boldsymbol{I} , \tag{3.34}$$

$$(\boldsymbol{\Lambda G})(\boldsymbol{G}^\dagger \boldsymbol{D}) = \boldsymbol{I} . \tag{3.35}$$

Thus Eqs. (3.28) and (3.29) represent an orthogonal wavelet decomposition. Equation (3.32) is the discrete equivalent of the condition $\int \psi(t)\, dt = 0$, making $\boldsymbol{g}$ a bandpass filter. Similarly, Eq. (3.33) makes $\boldsymbol{h}$ a low–pass filter. In the next section, we start with this (decimated) transform and derive an undecimated DWT.

3.2.2 The Undecimated *À Trous* Algorithm

The original transform given in [20] and [32] was the undecimated transform. This transform is translation–invariant. Following the development given in [50], we use this property here to derive the transform. The undecimated DWT will be denoted by $\widetilde{\boldsymbol{w}}$. Let T_m denote the translation of a signal by m

$$(T_m s)_n \triangleq s_{n-m} \ . \tag{3.36}$$

We define the undecimated DWT $\widetilde{\boldsymbol{w}}$ in terms of the decimated DWT $\boldsymbol{w}$ by

$$\widetilde{w}_n^i \triangleq [\widetilde{w}^i(s^0)]_n \triangleq [w^i(T_{-n}s^0)]_0 \ . \tag{3.37}$$

The relationship between the decimated and the undecimated transform is determined by the fact that the 0th element of a series is invariant under decimation so that w_n^i and $\widetilde{w}_n^i$ are related by

$$w_n^i = \widetilde{w}_{2^i n}^i \ . \tag{3.38}$$

Thus, sampling the undecimated transform $\widetilde{w}_{2^i n}^i$ every 2^i points produces exactly w_n^i. The filter sequence shown in Fig. 3.5 is used to calculate $\widetilde{w}$. The notation $D^i f$ indicates the decimation of the filter f by inserting $2^i - 1$ zeros between each pair of filter coefficients. The undecimated DWT is therefore given by

$$s^{i+1} = (D^i f) * s^i \ , \tag{3.39}$$

$$\widetilde{w}^i = (D^i g) * s^i \ . \tag{3.40}$$

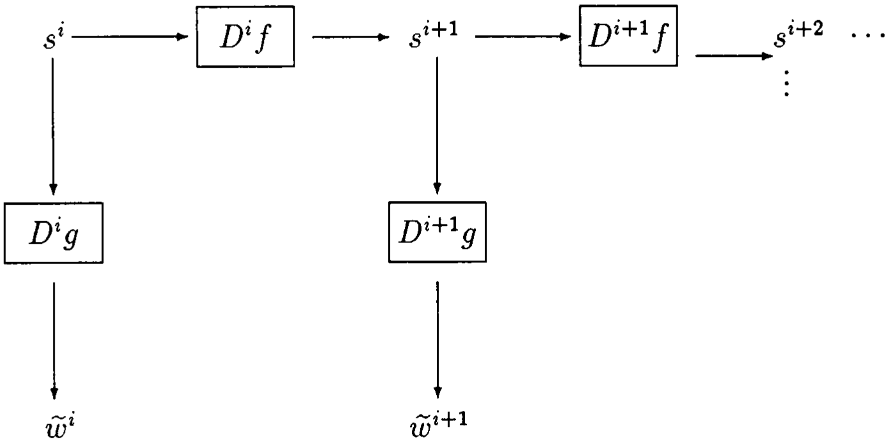


Figure 3.5: *The undecimated DWT.*

3.3 Comparison and Applications to Image Processing

Both algorithms presented above have been used successfully to deal with a variety of problems that arise in digital signal processing. Mallat's decomposition algorithm has been used for compression of 1– and 2–D signals [23, 36]. The *à trous* algorithm was used for analysis of speech and music signals [35].

Mallat's algorithm leads to an orthonormal decomposition. The strongest arguments in favor of orthonormality are the compact representation, mathematical elegance, ease of inversion, and good numerical properties. The wavelet representation has the same number of samples as the original signal we saw in Section 3.1.5. The mathematical elegance properties are not critical for the segmentation of images, nor is the ease of inversion. Numerically, the algorithm is stable and lends itself to efficient implementation. The major drawback of using this algorithm is that the scale of the output images is changed, shrinking each image by a factor of 2 in each direction at every stage. Analysis of these images thus becomes harder, while enlarging them to the original size using wavelets causes loss of resolution.

Mallat's algorithm was abandoned in favor of the *à trous* algorithm. The latter allows much more flexibility in filter design as the filters do not have to be orthogonal. Also, the scale of the image is unchanged throughout the algorithm. The insertion of zeros in lieu of decimation can be implemented implicitly in the algorithm so that the number of operations needed at each scale remains constant.

CHAPTER 4

Wavelets for Image Processing

One of the features that makes the WT diverse and applicable to many classes of problems in mathematics and signal processing is the flexibility available in using different families of wavelets for different purposes. The choice of the wavelet ψ to be used in the WT is essentially only restricted by Eqs. (2.3) and (2.4). In practice, one usually chooses ψ so that it is well concentrated in both time and frequency. This still leaves us with an infinite class of functions to choose from. A wavelet family can be constructed using any one of several methods, depending on the initial conditions specified. Most nontrivial wavelets cannot be expressed in closed form, they are usually given indirectly as a recursion. Thus their properties also have to be discovered indirectly. We begin this chapter by reviewing several mathematical properties of wavelets and their suitability to different applications, focusing specifically on the aspects that are needed for image processing. We then identify some desirable properties for the required wavelets, and derive and completely characterize a family of wavelets to be used later in the segmentation of range images. Frequency response analysis for the scaling functions and the wavelets is given. It is shown that scaling functions and wavelets act as low–pass and band–pass filters, respectively.

4.1 Properties of Wavelets

4.1.1 Orthogonality

This is probably the most fundamental property of a wavelet family, at least from a mathematical analysis viewpoint. An orthogonal wavelet family is a situation in which the mother wavelet is orthogonal to its dilations and translations. If the mother wavelet is denoted by $\psi(\mathrm{t})$, we have an orthogonal wavelet family if and only if

$$(\langle \psi, \psi(. - i) \rangle = \delta_i) \ .$$

The simplest such wavelet is the Haar function. The most commonly referred to family of orthogonal wavelets are the original Daubechies wavelets [15]. Orthogonal wavelets lead to ease of inversion and stable numerical properties, but lack flexibility in filter design. The wavelets derived later in this chapter are nonorthogonal.

4.1.2 Support

The support of a continuous function is the smallest closed set outside which the function vanishes identically. A function is said to *compactly supported* if it is nonzero over a closed set of points. When implementing the DWT on a computer, compactly supported wavelets give rise to finite impulse response (FIR) filters. Noncompactly supported wavelets, on the other hand, lead to infinite impulse (IIR) filters. FIR filters are more desirable as they lead to simpler and faster implementations. If the wavelets to be used are noncompactly supported but have a reasonably fast decay then the sequence can be truncated to give an FIR filter. Within this work we focus exclusively on wavelets that lead to FIR implementations. This is necessary for image processing algorithms to keep execution time within reasonable bounds.

4.1.3 Symmetry

All the wavelets encountered thus far in this work have been nonsymmetric except for the Haar basis. This follows Theorem 4.1 due to Daubechies [17].

Theorem 4.1: Let ϕ and ψ be the scaling function and wavelet associated with a multiresolution analysis, both real and compactly supported. If ψ has either a symmetry or antisymmetry axis, then ψ is the Haar function.

In other words, except for the Haar basis, all real orthonormal wavelet bases with compact support are asymmetric. The nonexistence of symmetric or antisymmetric real compactly supported wavelets is mirrored in subband coding theory [52], where it has been noted that symmetry is not compatible with the exact reconstruction property in subband filtering. The symmetry properties of a wavelet are important for image processing applications. Symmetric filters lead to identical results at image boundaries regardless of the direction in which the convolution was applied. In addition, asymmetric filters lead to more detectable quantization errors around the edges.

To gain symmetry, one must either drop the compact support or real wavelets condition. Symmetry is also possible if we use nonorthonormal wavelet bases. An example of this is the use of symmetric biorthogonal wavelet bases where each wavelet has a dual, and the pair satisfy the bi–orthogonality property.

4.1.4 Regularity

The concept of regularity [15] provides a measure of smoothness for wavelets and scaling functions. The regularity of the scaling function is defined as the maximum value of r such that

$$|\Phi(\Omega)| \leq \frac{c}{(1+|\Omega|)^{r+1}} \quad \Omega \in \mathbb{R} . \tag{4.1}$$

This, in turn, implies that $\phi(t)$ is m–times differentiable where $r \geq m$. The regularity or smoothness of $\phi(t)$ and $\psi(t)$ is determined by the decay of $\Phi(\Omega)$. More smoothness corresponds to better frequency localization of the filters. In addition, smooth basis functions are desired in applications to numerical differentiation. Wavelets chosen for detection of discontinuities in images and their derivatives should be continuously differentiable enough so as not to introduce any further singularities when applied to a signal.

4.1.5 Vanishing Moments

A wavelet $\psi(x)$ is said to have n vanishing moments if and only if for all positive integers $k < n$, it satisfies

$$\int_{-\infty}^{+\infty} x^k \psi(x)\, dx = 0 . \tag{4.2}$$

To see the impact of vanishing moments, let $\psi(x)$ further be a compactly supported wavelet. $\hat{\psi}(\omega)$ is n times continuously differentiable and one can derive from Eq. (4.2) above that $\hat{\psi}(\omega)$ has a zero of order n in $\omega = 0$. Thus, for any integer

$p < n$, $\hat{\psi}(\omega)$ can be factorized by

$$\hat{\psi}(\omega) = (i\omega)^p \hat{\psi}^1(\omega) \ . \tag{4.3}$$

In the spatial domain, this is equivalent to

$$\psi(x) = \frac{d^p \psi^1(\omega)}{d^p x} \ . \tag{4.4}$$

and the function $\psi^1(x)$ satisfies the wavelet admissibility condition. When wavelets are used in the detection and characterization of singularities, the number of vanishing moments comes into play. For characterization of Hölder (Lipschitz) continuous functions of order $\alpha \geq 1$, the wavelet must have a number of vanishing moments greater than α. Thus the number of vanishing moments limits the order of smoothness that can be characterized.

When the WT is taken of a 1–D signal or an image, the number of maxima of the transform often increase linearly with the number of moments of the wavelet. In order to minimize the amount of computations, we want to have the minimum number of maxima necessary to detect the interesting irregular behavior of the signal. This means we must choose a wavelet with as few vanishing moments as possible, but with enough moments to detect the Lipschitz exponents of highest order that we are interested in [41]. In image processing, we often want to detect discontinuities and peaks that have Lipschitz exponents smaller than 1. It is therefore sufficient to use a wavelet with only one vanishing moment.

4.2 Examples

In this section, two examples of wavelet families are given. From the conditions above on admissable wavelets, it is obvious that a strictly positive function cannot be a wavelet as some oscillations are needed to satisfy the $\int_{-\infty}^{\infty} \psi(x)\ dx = 0$ condition. Plane waves such as the trigonometric functions sin() and cos() oscillate infinitely without decay and are also unsuited for wavelet analysis. The wavelet families discussed next are the Haar wavelets and the Shannon wavelets. The Haar functions are the simplest example of orthonormal wavelet families, but are discontinuous in the time domain. The Shannon wavelets are on the other extreme, being discontinuous in frequency and hence spread out in time. Neither of these two is of much practical use, but they are given here to illustrate the main properties of the WT and for contrast with the wavelets to be derived subsequently for range image segmentation.

4.2.1 The Haar Wavelets

Let V_i denote the space of piecewise constant functions such that

$$V_i = \{f(t) \in L^2(\mathbb{R}); f \text{ is constant on } [2^i n, 2^i(n+1)] \forall n \in \mathbb{Z}\} .$$

These spaces satisfy the containment property

$$\cdots V_1 \subset V_0 \subset V_{-1} \cdots$$

Functions in these spaces also satisfy the scaling property

$$f(.) \in V_j \Longleftrightarrow f(2.) \in V_{j-1} .$$

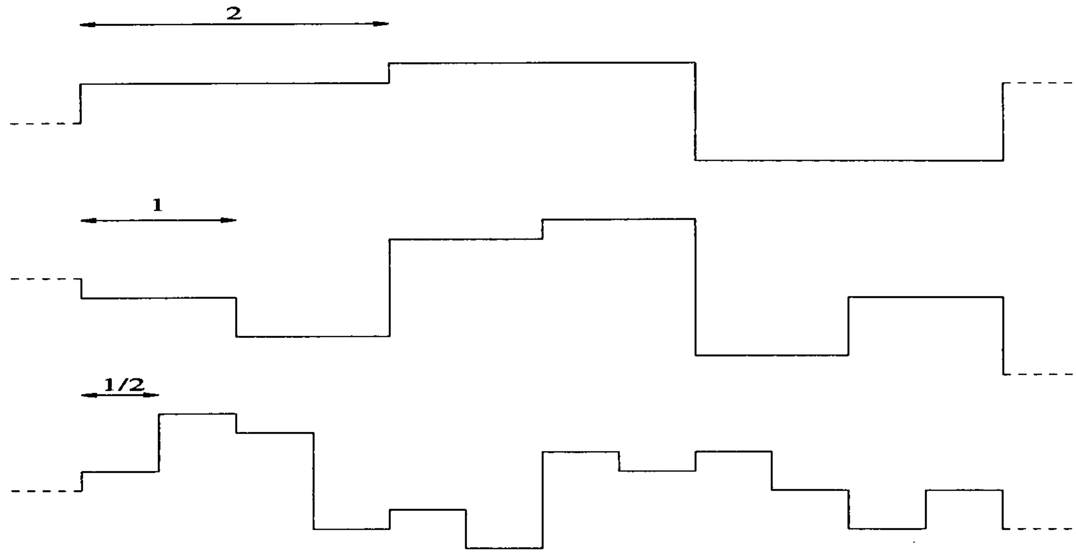


Figure 4.1: *Examples of functions in the subspaces V_i.*

Sample functions in these spaces are shown in Fig. 4.1.

The scaling function that corresponds to this multiresolution analysis is given by

$$\phi(t) = \begin{cases} 1, & \text{if } 0 \leq t < 1 \ , \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

with transform

$$\Phi(\Omega) = e^{-j\frac{\Omega}{2}} \frac{sin(\Omega/2)}{(\Omega/2)} \ . \tag{4.6}$$

The integer translations of the scaling function $\phi(t)$ form an orthonormal basis for V_0 since

$$\int \phi(t-n)\phi(t-m)dt = \delta_{n-m} \ .$$

If we use the recipe given in Section 2.4.2 we get

$$h_n = \sqrt{2}\int \phi(t)\overline{\phi(2t-n)}dt = \begin{cases} \sqrt{2}/2, & \text{if } n = 0, 1, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

Consequently,

$$\psi = \frac{1}{\sqrt{2}}\phi_{-1,0} - \frac{1}{\sqrt{2}}\phi_{-1,1} \ .$$

$$\psi(t) = \begin{cases} 1, & \text{if } 0 \le t < \frac{1}{2}, \\ -1, & \text{if } \frac{1}{2} \le t < 1, \\ 0, & \text{otherwise.} \end{cases} \tag{4.8}$$

The FT is given by

$$\Psi(\Omega) = e^{-j\frac{\Omega}{2}} \frac{\sin^2(\Omega/4)}{(\Omega/4)} \ . \tag{4.9}$$

The Haar scaling function and wavelet are shown in Fig. 4.2.

4.2.2 The Shannon Wavelets

These wavelets display properties that are the opposite to those of the Haar wavelets. They are discontinuous in the frequency domain and hence spread out in the time domain. As such, they are not compactly supported. Let V_0 be the space of bandlimited functions with support $(-\pi, \pi)$. Then by the Shannon sampling theorem, the functions

$$\phi(t-k) = \frac{\sin\ \pi(t-k)}{\pi(t-k)}, \quad k \in \mathbb{Z} \tag{4.10}$$

constitute an orthonormal basis for V_0. Any function $f(t) \in V_0$ can now be expressed as

$$f(t) = \sum_k f_k \frac{\sin\ \pi(t-k)}{\pi(t-k)} = \sum_k f_k \phi(t-k) \ . \tag{4.11}$$

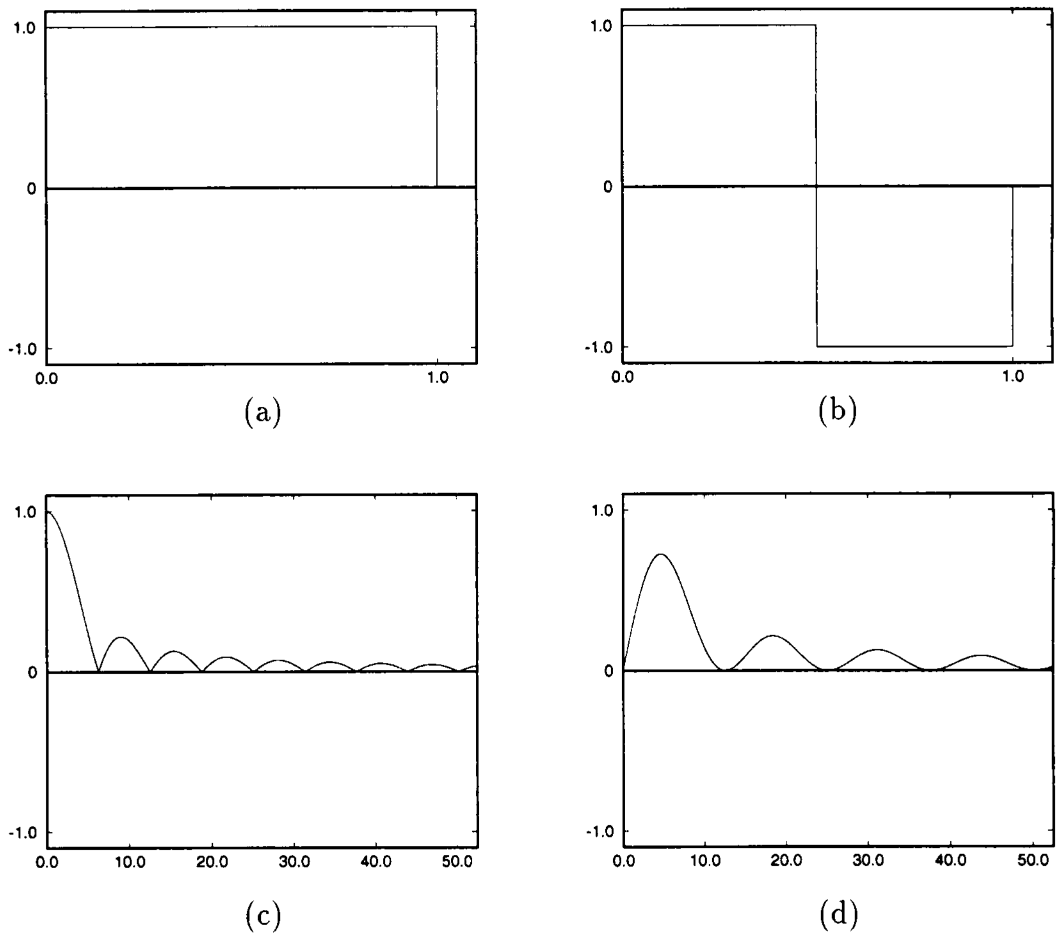


Figure 4.2: *The Haar wavelets: (a) Haar scaling function, (b) Haar wavelet function, (c) Fourier transform of Haar scaling function, and (d) Fourier transform of Haar wavelet function.*

In the frequency domain

$$\phi(t) = \frac{\sin\ \pi(t)}{\pi(t)} \longleftrightarrow \Phi(\Omega) = \begin{cases} 1 & \text{for } |\Omega| < \pi \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

The Shannon wavelet is given by

$$\psi(t) = \left(\frac{\sin \frac{\pi}{2}t}{\frac{\pi}{2}t}\right) \cos \frac{3\pi}{2}t \ , \tag{4.12}$$

$$\Psi(\Omega) = \begin{cases} 1 & \text{for } \pi < |\Omega| < 2\pi \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{4.13}$$

The Shannon scaling function and wavelet are shown in Fig. (4.3). Note that $\phi(t)$ and $\psi(t)$ are of infinite support. The wavelets are clearly not well localized in time, decaying only as fast as $1/t$.

4.3 Choice of Wavelets for Feature Extraction in Range Images

4.3.1 Desirable Properties of Candidate Wavelets

As we have seen, the WT is unique in that it does not specify the basis functions to be used except that the mother wavelet chosen should satisfy Eqs. (2.3) and (2.4). This leaves numerous families of functions to choose from. Indeed, there are several mathematical tools that can give rise to wavelet families, including recursion [18, 53], spline equations [11, 44], and fractal analysis [44]. For the purpose of edge detection in images, we impose the following conditions on any candidate wavelet:

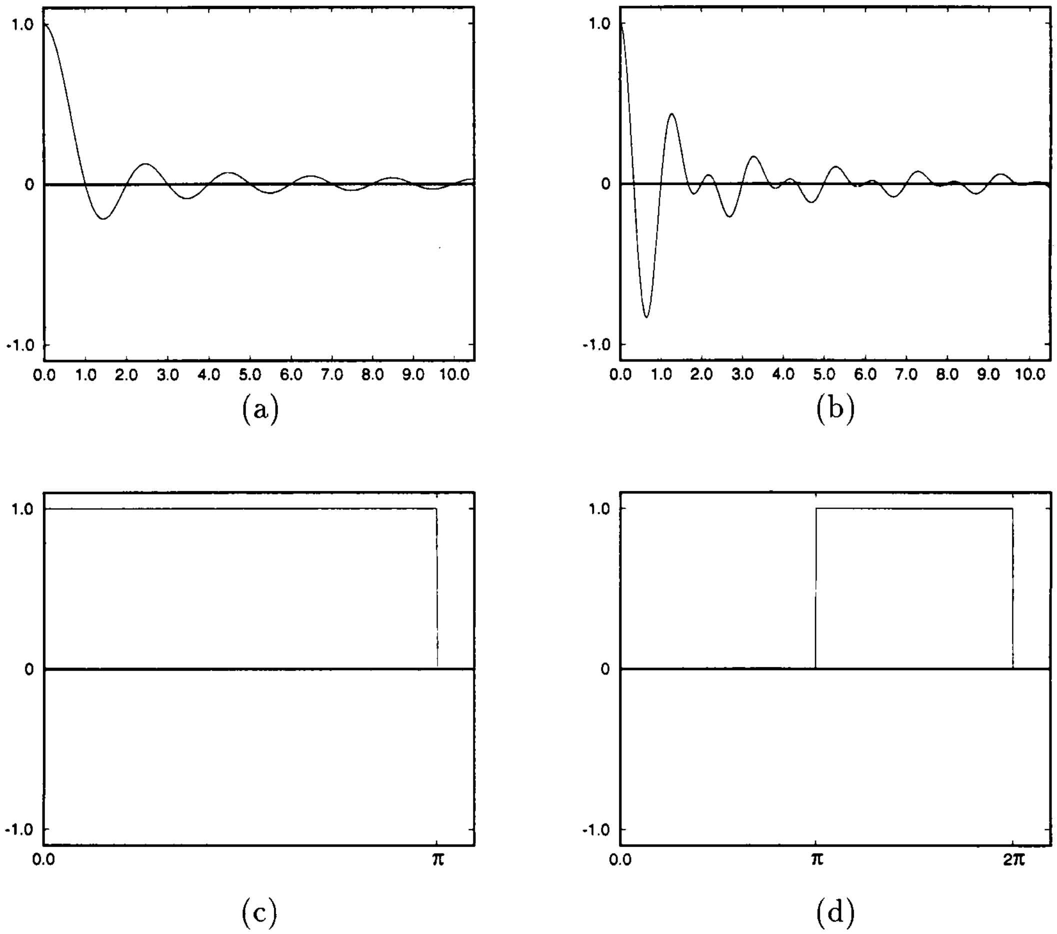


Figure 4.3: *Shannon wavelets: (a) Shannon scaling function, (b) Shannon wavelet function, (c) Fourier transform of Shannon scaling function, and (d) Fourier transform of Shannon wavelet function.*

- Compact support which gives rise to finite impulse response (FIR) filters leading to simpler and faster implementations.

- Low number of coefficients which relates to the compact support property above, but is also necessary for minimizing edge interaction. The scale of the original image and the nature of its components determine the minimum expected inter–edge distance. The number of coefficients also affects the time taken to calculate one pass of the transform.

- Differentiability, since the wavelets chosen for detection of discontinuites in images and their derivatives should be continuously differentiable enough so as not to introduce any further singularities when applied to a signal.

4.3.2 Proposed Wavelet Family

A scaling function is derived from a solution to the two–scale difference equation as given by Eq. (2.9):

$$\phi(x) = \sum c_k \phi(2x - k) \ , \qquad (4.14)$$

where $\phi(x)$ is normalized by $\int \phi(x)\ dx = 1$ as given before. This method allows construction of wavelets with a given number of coefficients subject to these constraints. To find the coefficients c_k, we first integrate the above equation yielding $\sum c_k = 2$. We define a wavelet from $\phi(x)$ by

$$\psi(x) = \sum (-1)^k c_{1-k} \phi(2x - k)$$

The low–pass filter coefficients are given by $h_n = c_n/2$, while the bandpass filter coefficients are given by

$$g_n = (-1)^{n-1} h(1-n) \; . \tag{4.15}$$

Equation (4.14) can be solved for any number of the coefficients c_n. For the purpose at hand, we decided to use four coefficients. As we are mostly interested in analysis at the first three scales, the length of the filters will be 4, 7, and 10, since zeros are inserted between the coefficients at each scale. The rationale behind the choice for the number of coefficients can be summarized as follows:

- The filter lengths obtained, 4, 7, and 10, are compatible with regard to the minimization of the edge interaction requirement above. The images used in this work were all 128 × 128. Choosing a five–coefficient wavelet would lead to filter lengths of 5, 9, and 13. A filter length of 13 spans approximately 10% of the image leading to a significant smearing effect due to interaction of neighboring features. A three–coefficient wavelet would yield filter lengths of 3, 5, and 7. These are also appropriate choices but were dropped due to noise and performance criteria below.

- Optimum results for analysis of range images have mostly been reported in the literature with mask sizes of 5 × 5, 7 × 7, and 9 × 9 [3, 4, 6, 21, 47]. Several reasons have been cited for these sizes. A 3 × 3 mask attaches strong emphasis to the center pixel and does not provide enough immunity to noise. As the mask gets bigger, it provides more tolerance to noise, but edge–localization becomes increasingly poor and edges that are close begin to interact, producing a combined edge response. The filter lengths obtained with our four–coefficient wavelet 4, 7, and 10 are closest possible to the optimum mask sizes (for example a 3– coefficient wavelet gives rise to filter lengths 3, 5, and 7, whereas a 5–coefficient wavelet gives rise to filter lengths of 5, 9, and 13 as indicated before).

- A complete characterization of four–coefficient scaling functions and wavelets already existed [13]. This study was used as a basis for the wavelets used in this work. It was extended by imposing conditions on the general class of four–coefficient wavelets. Although constructing wavelets from Eq. 4.14 can be done iteratively for any number of coefficients using a computer, complete characterization of the mathematical properties for the resulting wavelets is an arduous task that requires extensive knowledge of wavelets. This is especially so because we do not have closed form expressions for any of the scaling functions or the wavelets. The objective of this work is to find methods of application of the WT to solve the range image segmentation problem. The techniques and algorithms developed herein can be used with different classes of wavelets without modification, and they constitute a basis for future work on wavelets and range image segmentation.

With four coefficients, the two–scale dilation Eq. 4.14 now becomes

$$\phi(x) = \sum_{k=0}^{3} c_k \phi(2x - k) \ . \tag{4.16}$$

The c_k's also satisfy $\sum c_{2k} = \sum c_{2k+1} = 1$, as this is a necessary condition for the existence of a multiresolution analysis, so

$$c_0 + c_2 = 1$$

$$c_1 + c_3 = 1 \ .$$

The characterization of continuous four–coefficient scaling functions and wavelets was studied by Colella and Heil [13] , varying c_0 and c_3 as independent parameters. It was shown that there are no integrable solutions to Eq. (4.16) on or outside the ellipse

$$c_0^2 + c_3^2 - c_0 - c_3 + c_0 c_3 = 1 \ ,$$

except the point $(c_0, c_3) = (1, 1)$. It was also shown that there is an integrable solution to Eq. (4.16) at every point on and inside the circle

$$(c_0 - 1/2)^2 + (c_3 - 1/2)^2 = \sqrt{2}/2 \ .$$

This circle is referred to as the *circle of orthogonality.* The locii of these objects in the (c_0, c_3) plane are shown in Fig. 4.4.

The triangle in Fig. 4.4 gives an approximation to the region of points resulting in continuous scaling functions [13]. The point $(c_0, c_3) = (1, 0)$ gives the scaling function corresponding to the Haar system reviewed in Section 4.2.1. The points $(0, 1)$ and $(0, 0)$ are translates of this system, while $(1, 1)$ gives a stretched version.

The line $c_0 + c_3 = 1/2$ identifies a necessary condition for a scaling function to be differentiable. The differentiable solutions are indicated by the solid line segment in Fig. 4.4. The intersection of the line and the circle gives the Daubechies scaling function D_4 [15] which is differentiable almost everywhere. The domain for

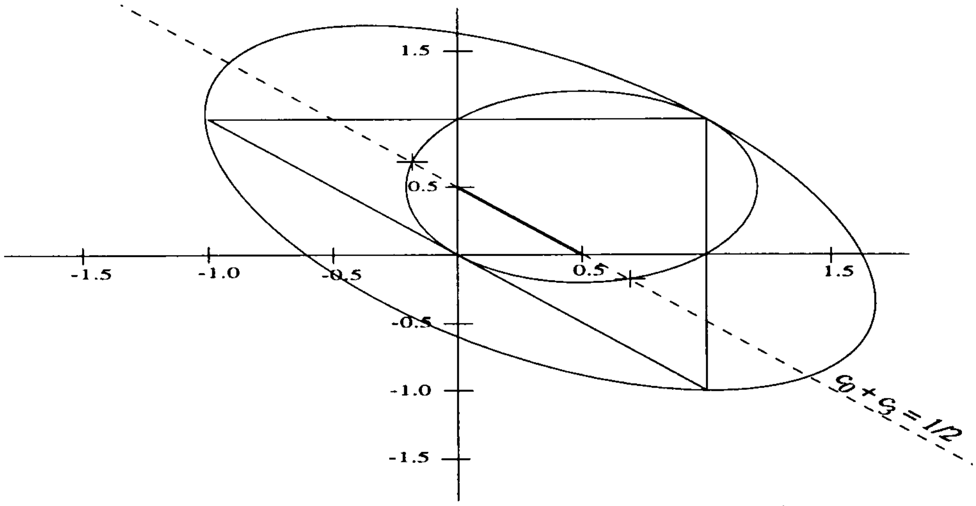


Figure 4.4: *Characterization of four–coefficient scaling functions: The* (c_0, c_3) *plane showing the ellipse* $c_0^2 + c_3^2 - c_0 - c_3 + c_0c_3 = 1$*, the circle* $(c_0 - 1/2)^2 + (c_3 + 1/2)^2 = \sqrt{2}/2$*, the line* $c_0 + c_3 = 1/2$*, and a triangle. The crosses at the intersection of the line and circle are the Daubechies* D_4 *scaling function [15].*

---

our solution lies in the first quadrant. These scaling functions are not orthogonal because there are no orthogonal four–coefficient functions that are differentiable everywhere. Further, symmetrical scaling functions lie on $c_0 = c_3$. The proposed family therefore lies on this line. The support of the possible scaling functions lies in $[0, 3]$.

In order to define the scaling function, we start with Eq. (4.16).

$$\phi(x) = c_0\phi(2x) + c_1\phi(2x-1) + c_2\phi(2x-2) + c_3\phi(2x-3) \tag{4.17}$$

For the function $\phi(x)$ to be continuous, we must have $\phi(0) = \phi(3) = 0$. Using

Eq. (4.17) with $x = 1, 2$ we get

$$\begin{pmatrix} \phi(1) \\ \phi(2) \end{pmatrix} = \begin{pmatrix} c_1 & c_0 \\ c_3 & c_2 \end{pmatrix} \begin{pmatrix} \phi(1) \\ \phi(2) \end{pmatrix} . \tag{4.18}$$

Define the matrix M

$$M = \begin{pmatrix} c_1 & c_0 \\ c_3 & c_2 \end{pmatrix} . \tag{4.19}$$

The eigenvalues of M are 1 and $1 - c_0 - c_3$. The eigenvector associated with the eigenvalue 1 yields

$$\phi(1) = c_0 \ , \tag{4.20}$$

$$\phi(2) = c_3 \ . \tag{4.21}$$

The other eigenvalue $1 - c_0 - c_3$ leads to the derivatives $\phi'(1)$ and $\phi'(2)$.

Using the values for $\phi(1)$ and $\phi(2)$ and keeping in mind $\phi(0) = \phi(3) = 0$, we can now use the recursion in Eq. (4.16) to calculate $\phi(x)$ at the half–integers, the quarter–integers, and ultimately at all the dyadic points $x = k/2^j$. Possible choices for the scaling function and wavelet coefficients are given in Table 4.1. An all–pass filter L is defined by its frequency domain response

$$L(\omega) = \frac{1 + |H(\omega)|^2}{2} \ .$$

The L filter is used as an all–pass filter and is applied after each convolution with H or G. Figure 4.5 shows the the shape of the resulting scaling functions for the coefficients given in Table 4.1, while Fig. 4.6 gives the frequency response.

Table 4.1: Possible choices of coefficients for the derived wavelets.

$c_0 = c_3 = 0.05$

n	$h(n)$	$g(n)$	$l(n)$
-3			0.0003
-2			0.0119
-1	0.025	0.025	0.1247
0	0.475	-0.475	0.7262
1	0.475	0.475	0.1247
2	0.025	-0.025	0.0119
3			0.0003

$c_0 = c_3 = 0.10$

n	$h(n)$	$g(n)$	$l(n)$
-3			0.0013
-2			0.0225
-1	0.050	0.05	0.1237
0	0.450	-0.45	0.7050
1	0.450	0.45	0.1237
2	0.050	-0.05	0.0225
3			0.0013

$c_0 = c_3 = 0.20$

n	$h(n)$	$g(n)$	$l(n)$
-3			0.0050
-2			0.0400
-1	0.100	0.100	0.1200
0	0.400	-0.400	0.6700
1	0.400	0.400	0.1200
2	0.100	-0.100	0.0400
3			0.0050

$c_0 = c_3 = 0.25$

n	$h(n)$	$g(n)$	$l(n)$
-3			0.0078
-2			0.0469
-1	0.125	0.125	0.1172
0	0.375	-0.375	0.6562
1	0.375	0.375	0.1172
2	0.125	-0.125	0.0469
3			0.0078

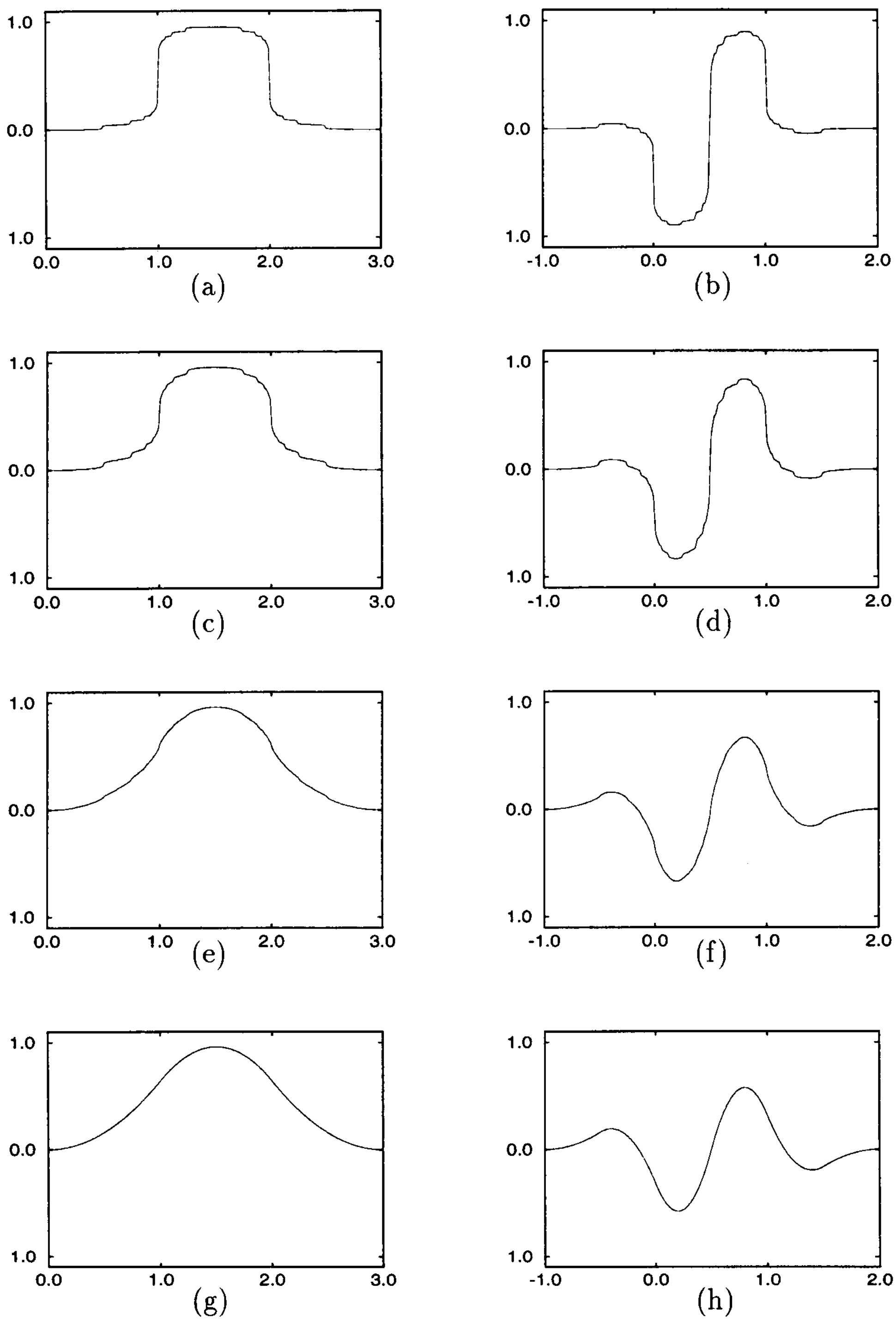


Figure 4.5: *Chosen four–coefficient symmetric scaling functions and their associated wavelets: (a) and (b):* $c_0 = c_3 = 0.05$, *(c) and (d):* $c_0 = c_3 = 0.10$, *(e) and (f):* $c_0 = c_3 = 0.20$, *and (g) and (h):* $c_0 = c_3 = 0.25$.

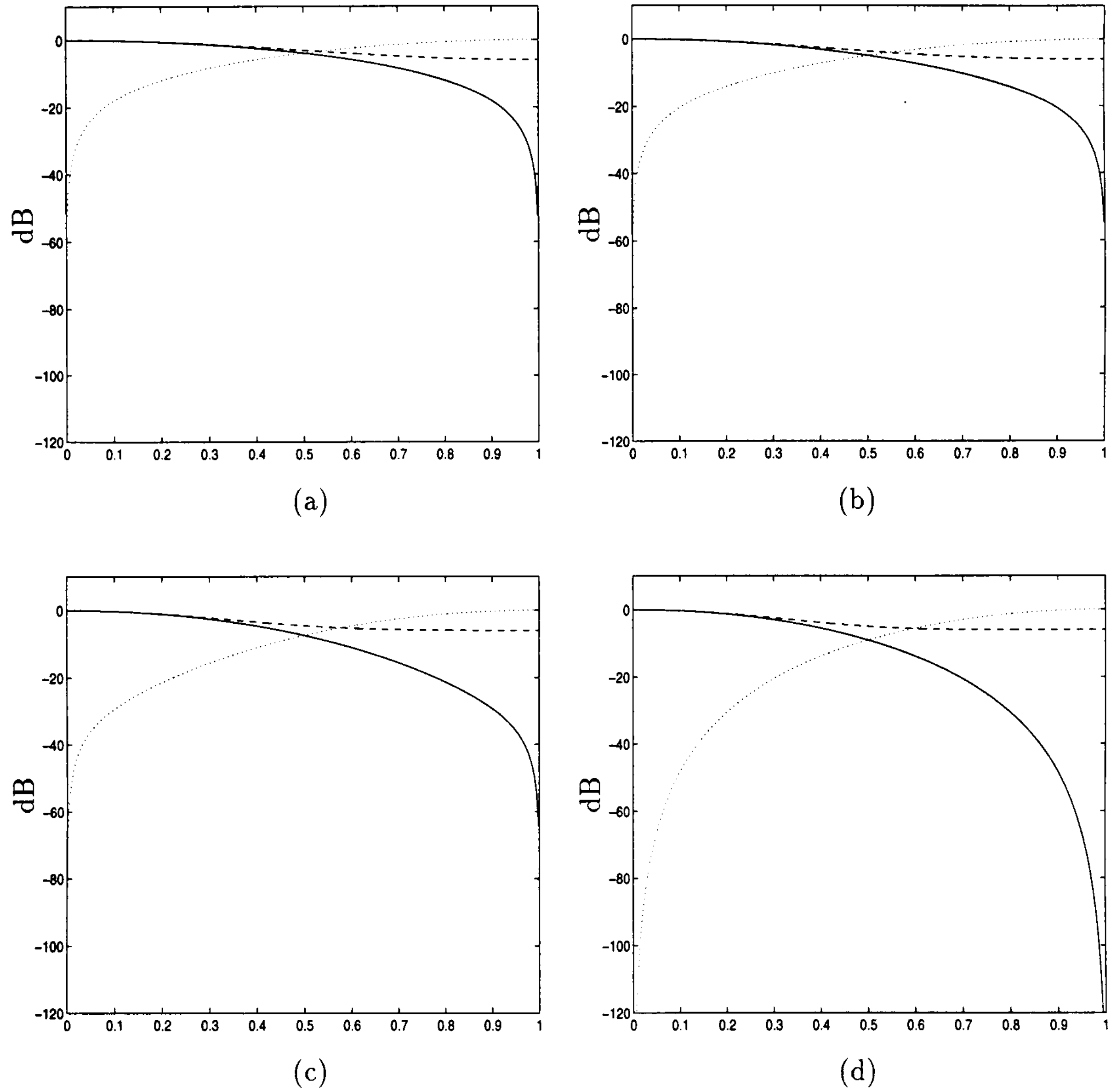


Figure 4.6: *Magnitude response (dB) versus normalized frequency of chosen scaling functions (solid line), wavelets (dotted line), and all–pass filters (dashed line) for the cases (a):* $c_0 = c_3 = 0.05$, *(b):* $c_0 = c_3 = 0.10$, *(c):* $c_0 = c_3 = 0.20$, *and (d):* $c_0 = c_3 = 0.25$.

CHAPTER 5

Derivative Estimation for Range Images Using Wavelets

In studying discrete signals, often one cannot analyze the information content by direct inspection of the values at hand. Instead, we are usually more interested in the variations across the signal. In computer vision, we often need to isolate the boundaries between objects in a scene. In range images, these boundaries fall into two broad categories, digital step edges and roof or trough edges. Step edges are represented by a jump in the gray–level intensity and indicate a discontinuity between two objects in the scene or an object and its background. The term roof edges is used within this work for any boundary between two surfaces where the gray–level intensity values do not have jumps between them.

Segmentation of images in computer vision can be either edge–based or area–based. In edge–based segmentation we attempt to find and isolate the different types of edges then aggregate them into an edge map for the scene. This edge map defines bounded areas or patches for higher–level recognition steps. In area–based segmentation, we start with a number of *seeds* and grow them into areas that eventually meet and form boundaries between them. This work is of the former type, where we try to locate different types of edges by using a model of the differential characteristics of these edges. Detecting edges in range images is done by finding the *partial derivatives* of an image. There is no strict definition of derivatives for a discrete signal, and what we actually calculate are different (discrete) approximations of these derivatives. The main methods used in image processing for extracting derivatives from discrete images include

- Surface fitting, in which we assume that the underlying signal is a polynomial. We thus apply local windows such as 3×3 or 5×5 to an image and fit polynomials to it subject to various error–minimization constraints. Once we do the surface fitting, we can calculate the derivatives of the center of the window from the coefficients of the fitted polynomials. This is repeated throughout the image.

- Convolution, in which we use an auxiliary function for which the derivative exists and is known. By convolving the derivative of the function locally with the image, we get an estimate of the derivative of the image convolved with the original function. This is the basis of the method used within this work for estimating derivatives.

Using the WT for detecting variations in signals gives us a differentation–smoothing operator with built–in multiscale analysis.

5.1 Multiscale Derivative Approximations for 1–D Signals

Let $\phi(x)$ be a smoothing function such that its integral is equal to 1 and the function decays to 0 at infinity. Further, assume that $\phi(x)$ is n–times differentiable. We define a function $\psi^i(x)$ as the i^{th} derivative ($i < n$) of the smoothing function $\phi(x)$.

$$\psi^i(x) = \frac{d^i}{dx^i}\phi(x)$$

From the definition of $\psi^i(x)$ and the conditions on $\phi(x)$

$$\int_{-\infty}^{+\infty} \psi^i(x)dx = 0 \ .$$

The function $\psi^i(x)$ satisfies Eqs. (2.3) and (2.4) and can be considered a wavelet. We can write the CWT of a signal f with respect to this function as the convolution of f with the dilated wavelet

$$W_s^i f(x) = f * \psi_s^i(x) \ , \tag{5.1}$$

from which we derive

$$W_s^i f(x) = f * (s^i \frac{d^i \phi_s}{dx^i})(x) = s^i \frac{d^i}{dx^i}(f * \phi_s)(x) \ . \tag{5.2}$$

Thus $W_s^i f(x)$, the WT of f at scale s, is the i–th derivative of the signal smoothed at the scale s. This is a generalization of the convolution–based edge detectors. For example, if we use the Gaussian as the smoothing function, then detecting extrema for $i = 1$ corresponds to a multiscale Canny edge detector [10]. Detecting zero–crossings with $i = 2$ gives us a Marr–Hildreth edge detector [43]. Many other mask edge detectors are also examples of this formula.

5.2 Extension to Images

5.2.1 2–D Directional Derivatives Approximations

The derivations above can be extended to 2–D signals or images. However, an additional complexity in this case is the 2–D convolution. In this work, convolution is accomplished by using separable wavelets and 2–D convolutions. Let $\phi(x, y)$ be a 2–D smoothing function such that

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \phi(x,y)\, dx\, dy = 1$$

and

$$\lim_{x\to\pm\infty} \phi(x,y) = \lim_{y\to\pm\infty} \phi(x,y) = 0 \ .$$

We define two wavelet functions $\psi^1(x, y)$ and $\psi^2(x, y)$ such that

$$\psi^1(x,y) = \frac{\partial\phi(x,y)}{\partial x} \tag{5.3}$$

and

$$\psi^2(x,y) = \frac{\partial\phi(x,y)}{\partial y} \ . \tag{5.4}$$

We define the dilations of the wavelets by

$$\psi^1_s(x,y) = \frac{1}{s^2}\psi^1(\frac{x}{s},\frac{y}{s})$$

and

$$\psi^2_s(x,y) = \frac{1}{s^2}\psi^2(\frac{x}{s},\frac{y}{s}) \ .$$

The WT of $f(x, y) \in \mathbb{L}^2(\mathbb{R}^2)$ at the scale s has two components defined by [42]

$$W^1_s f(x,y) = f * \psi^1_s(x,y)$$

and

$$W^2_s f(x,y) = f * \psi^2_s(x,y) \ ,$$

from which, by convolution property, we get [42]

$$\begin{pmatrix} W^1_s f(x,y) \\ W^2_s f(x,y) \end{pmatrix} = s\begin{pmatrix} \frac{\partial}{\partial x}(f*\phi_s)(x,y) \\ \frac{\partial}{\partial y}(f*\phi_s)(x,y) \end{pmatrix} = s\vec{\nabla}(f*\phi_s)(x,y) \ .$$

The above can also be extended to higher–order derivatives.

5.2.2 Magnitude and Angle of 2–D Wavelet Transform

The magnitude of the 2–D WT at the scale s is given by

$$M_s f(x,y) = \sqrt{|W_s^1 f(x,y)|^2 + |W_s^2 f(x,y)|^2} \tag{5.5}$$

This is called the magnitude or modulus of the 2–D WT. If the wavelets are defined from a smoothing function as given by Eqs. (5.3) and (5.4), then it is proportional to the magnitude of the vector $s\vec{\nabla}(f * \phi_s)(x,y)$.

The angle of the 2–D WT is defined by

$$A_s f(x,y) = \tan^{-1}\left(\frac{W_s^2 f(x,y)}{W_s^1 f(x,y)}\right) . \tag{5.6}$$

This is the angle between the vector $\vec{\nabla}(f * \phi_s)(x,y)$ and the horizontal. It gives the direction in which the signal has the sharpest variation.

5.3 Modulus Maxima of the Wavelet Transform

Before we proceed to discuss wavelet maxima, it is necessary to point out exactly what we mean by the local maxima of the WT magnitude. By this we are referring to a local maximum with a positive amplitude or a local minimum with a negative amplitude. This is to be distinguished from a local minimum with a positive amplitude or a local maximum with a negative amplitude which are local maxima of amplitude, not magnitude. These are called phantom edges [12]. Within this work, we shall use local maxima to refer to local maxima of magnitude. These maxima occur at the points where the modulus is locally maximum in the direction given by the angle at that point. Examples of the modulus and the angle images are shown in Figs. 5.1 and 5.2 at different scales.

5.3.1 Maxima and Zero–Crossing Detection for Edges

Having calculated the WT of a signal at a given scale, we are in a position to analyze the information it conveys. We start with $W_s^1 f(x)$ and $W_s^2 f(x)$, the first and second derivative of the signal smoothed at the scale s. The inflection points of $f * \phi_s(x)$ give local extrema in $W_s^1 f(x)$ and zero–crossings in $W_s^2 f(x)$. In the particular case when $\phi(x)$ is a Guassian, extrema detection of $W_s^1 f(x)$ is equivalent to a multiscale Canny edge detector, whereas zero–crossing detection is equivalent to a Marr–Hildreth edge detector.

Zero–crossings and local maxima detection have both been used extensively in the literature for analyzing digital images. Each approach has advantages. As mentioned above, an inflection point of $f * \phi_s(x)$ can be either a maximum or a minimum of the absolute value of its first derivative. The maxima of the absolute value of the first derivative are sharp variation points of $f * \phi_s(x)$, whereas the minima correspond to slow variations. A first derivative operator can distinguish between the two, whereas a second derivative operator gives a zero crossing at both types of points. Thus zero–crossings give position information but do not differentiate small amplitude fluctuations from important discontinuities. On the other hand, the detection of zero–crossings eliminates the need for thresholding which is needed for detecting maxima. Thus the zero–crossings detection procedure lessens the number of parameters needed for the segmentation algorithm.

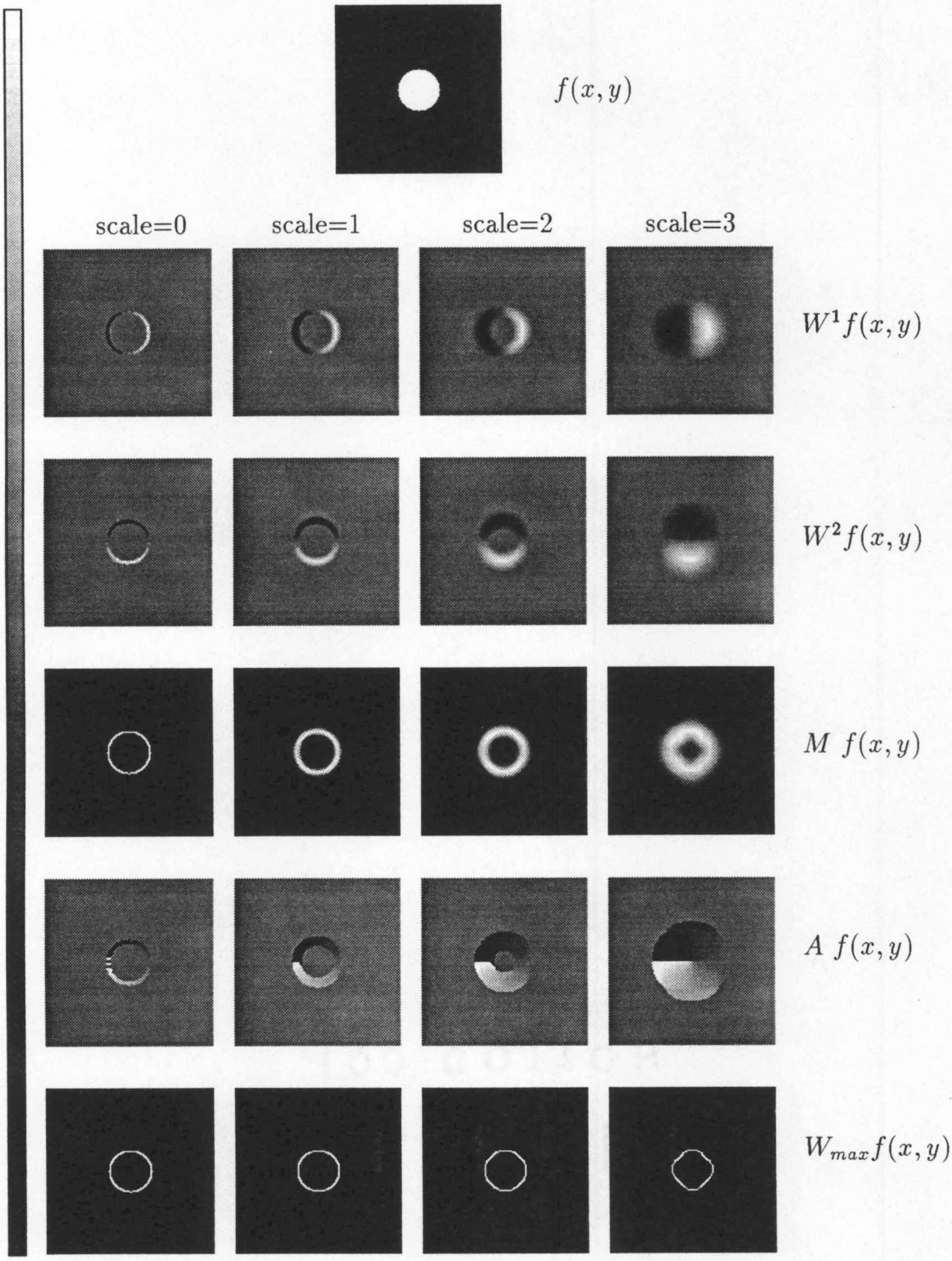


Figure 5.1: *The 2-D DWT of a synthetic image.*

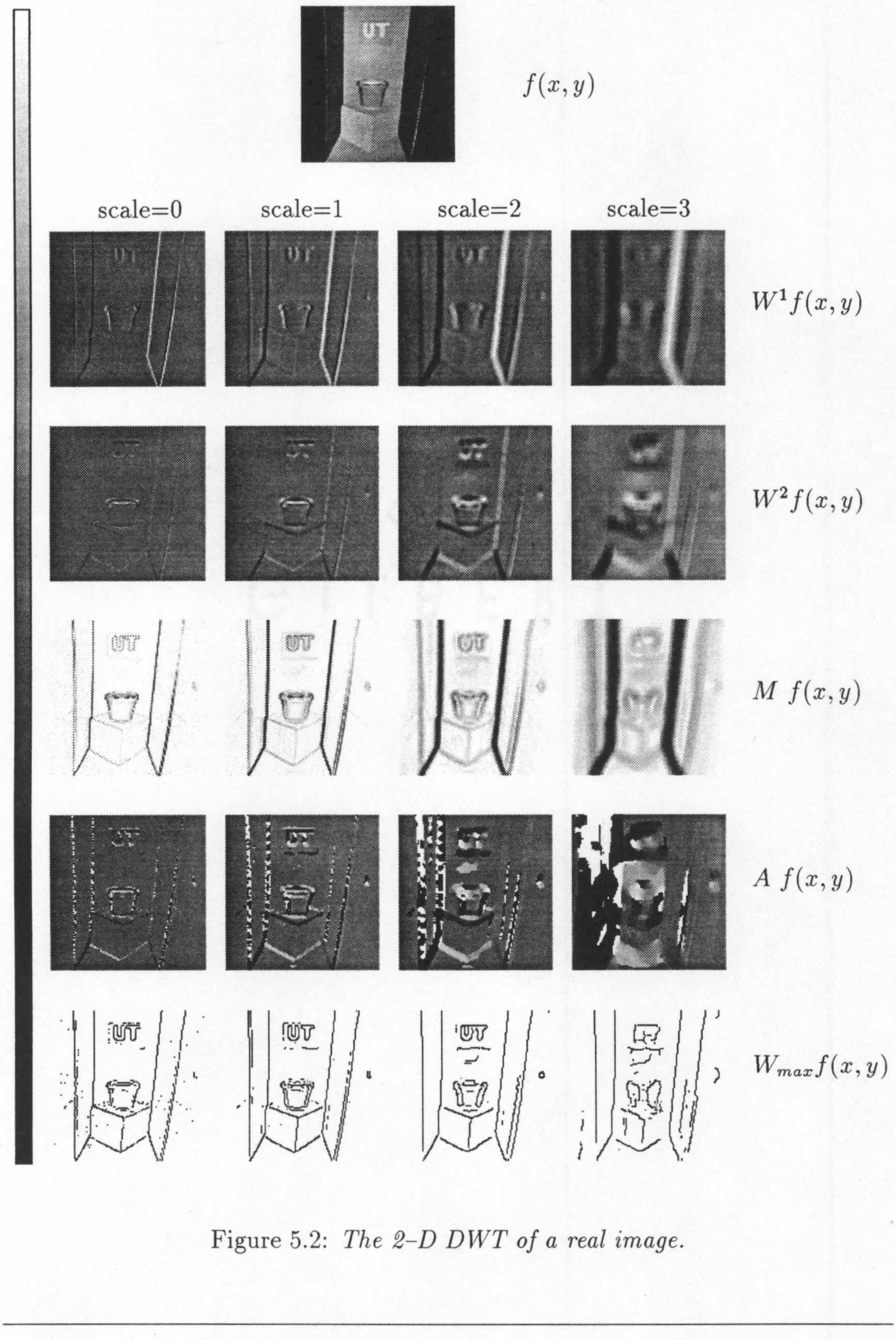


Figure 5.2: *The 2–D DWT of a real image.*

Detecting different types of edges is best done by making use of both approaches and combining the findings.

5.3.2 Step Edge Detection

We define a step edge as a sharp transition in the intensity level between two surfaces. Ideally, this transition would occur from one pixel immediately to the next one, but sometimes it takes place over two or three pixels. A typical step edge and its first and second derivatives are shown in Fig. 5.3.

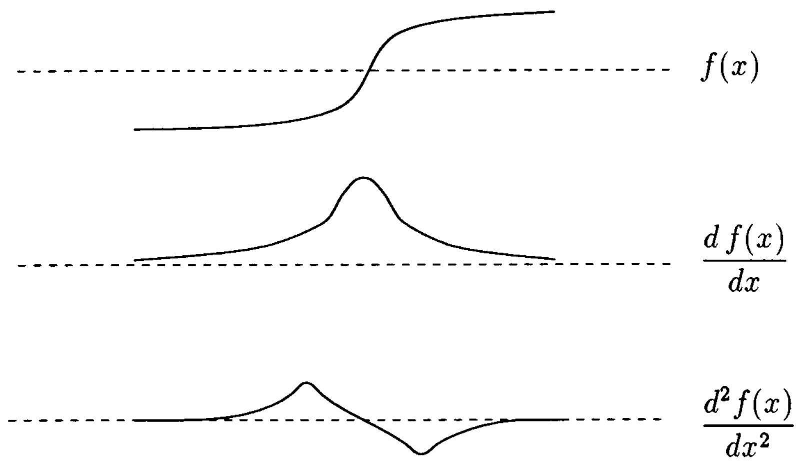


Figure 5.3: *A typical step edge and its first two derivatives.*

As can be seen from the figure, a step edge can be characterized by a maxima at the first derivative or a zero–crossing at the second derivative. These are the basis for the Canny and the Marr–Hildreth edge detectors, respectively.

5.3.3 Roof Edge Detection

A roof edge occurs whenever two surfaces meet without jumps in the intensity level values. Some types of roof edges are shown in Fig. 5.4.

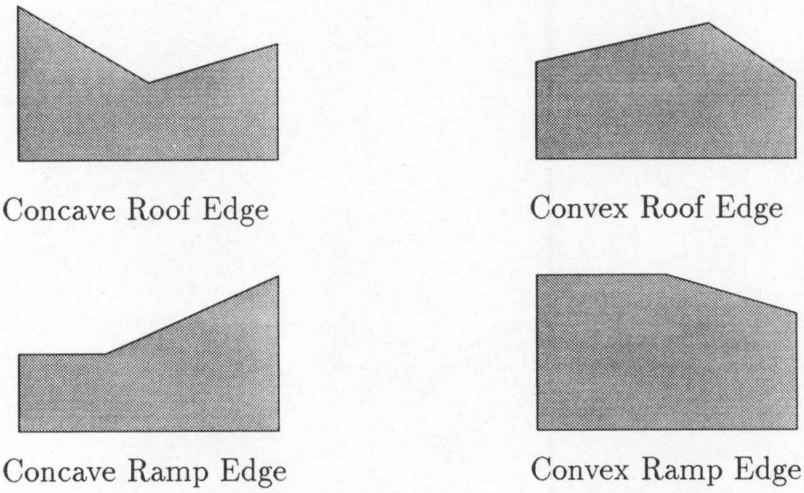


Figure 5.4: *Example of roof edges.*

Roof edges are often classified and labelled according to topographic elements such as peaks, pits, ridges, valleys, and saddles [31]. A typical roof edge with its first and second derivatives is shown in Fig. 5.5. Characterization and detection of roof edges is considerably more difficult than for step edges. As can be seen from Fig. 5.5, a roof edge usually gives rise to a zero–crossing in the first derivative, even if it has a different profile, as in Fig. 5.4. The second derivative usually exhibits spike–like waves on both sides of the edge, with a nonzero value in the middle. The easiest approach based on the behavior of the derivatives near the roof edge is looking for points where the first derivative is zero and the second derivative is

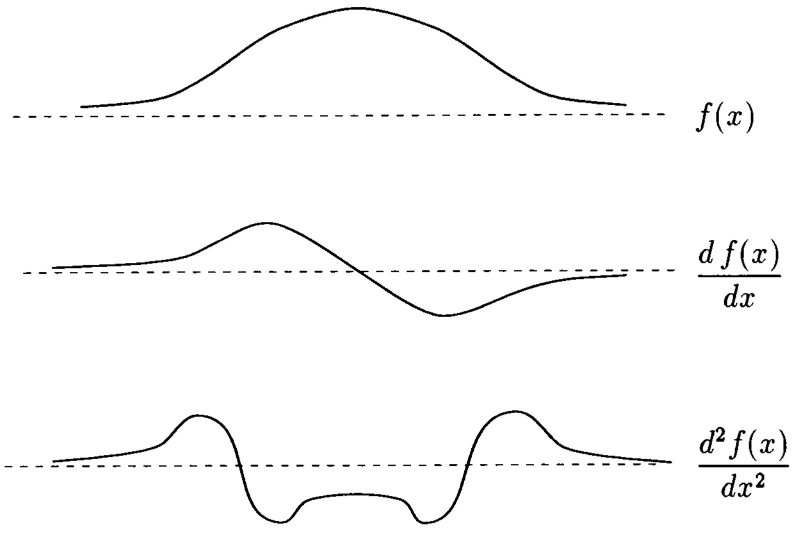


Figure 5.5: *Profile of a sample roof edge and its first two derivatives.*

nonzero. Assuming the roof edge is sufficiently convex or concave, this approach would locate most roof edges. In practice, the second derivative estimation is prone to noise and other errors, limiting the success of this procedure.

5.4 Range Images and Angle of 2–D Wavelet Transform

The WT angle image can be useful in detecting roof edges. It maps the range image into an intensity image so that an entire surface in the range image has a fixed intensity level and surfaces with different orientations have different intensity levels. This is also known as the surface normal image [4] in range image analysis. However, we have arrived at it here using the WT. In theory, detecting roof edges gets reduced to detecting step edges. Examples of this are included in the results shown later. There are, however, some drawbacks to this technique. It is highly sensitive to noise, and tends to introduce false edges in the presence of surfaces with sharp curvatures. These problems are apparent in the lower part of the knot in the ESCHERKNOT image.

5.5 Practical Considerations

When applying convolution, there are many practical aspects that need to be taken into consideration. The following are some of the issues that were dealt with.

Border of Image

The issue of convolution at the borders of images was dealt with by assuming symmetry around each of the four sides of the image. In other words, if we let f be a discrete 1–D signal with $N+1$ samples

$$f = (\theta_n)_{0 \leq n \leq N} \ ,$$

then we assume symmetry around $n = 0$ and $n = N$

$$\theta_n = \begin{cases} \theta_{-n} & \text{if } -N < n < 0 \ , \\ \theta_{2N-n} & \text{if } N < n < 2N \ . \end{cases}$$

Thus the signal can be repeated infinitely in both directions. Applying this procedure to images was found to yield the least visually perceptible irregularities at the boundaries of the resulting images.

Shift–Compensation

Whenever a convolution filter is applied to a signal, the result is displaced by half the length of the convolution filter. Repeated convolutions, as with the *à trous* algorithm lead to features that are displaced from their original positions. These displacements can be adjusted for by displacing the result of the convolution back by half the length of the convolution filter. However, if the length of the filter is odd, this necessitates shifting to within subpixel position, which is harder to achieve. In this work, we use even–length filters exclusively to allow for the shift–compensation to work at pixel level and without needing subpixel accuracy.

CHAPTER 6

Experimental Results

This chapter presents two algorithms based on the analysis given so far for the segmentation of range images. The objective is to generate an edge map from a range image that is an accurate outline of all the edges in the scene with minimum interference or noise. These edge maps are then processed for producing closed contours and labeled to give a final segmented image. The algorithms presented use the wavelets derived in Chapter 4 for edge detection. The first algorithm uses one pass of the DWT for calculating the WT modulus maxima. The result is an edge map that describes the large–scale features in the image. The second algorithm uses the angle image generated from the first pass and applies to it the DWT a second time. This generates an edge map indicating surface discontinuities in the range image. These edge maps are then fused to produce a complete binary edge map ready for labeling.

This chapter presents extensive range image segmentation results using a wide variety of range images comprising synthetic and real range data. In addition, performance of the algorithm under noise is analyzed. The computational complexity of the technique used is calculated, and approximate times for execution on a Sun SPARCstation 10 are given.

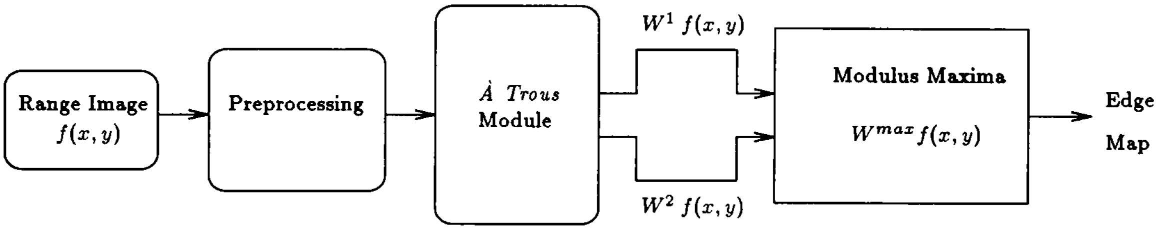


Figure 6.1: *Block diagram for the WT modulus maxima tracing algorithm.*

6.1 Wavelet Transform Modulus Maxima Tracing Algorithm

This algorithm applies the DWT and extracts wavelet maxima directly from the calculated modulus image and angle images. Figure 6.1 shows a block diagram of the modules used within the algorithm. Each module represents several algorithmic steps performed on the image.

6.1.1 Preprocessing

The first module, labeled *preprocessing*, consists of steps taken to precondition the raw image and remove noise present in it. The nature of noise depends on the environment in which the scanner is being used and on the scanner itself. In this work, we focused on the elimination of two types of noise present in most data. The first type is spike noise, which manifests itself as very high values distributed sparsely in the image. This noise can be removed by statistical filtering of the data, which measures the variance of each pixel from a global or a local statistical measure, such as the average, and conditionally applies median filtering if the range value is very deviant from its neighbors.

The other type of noise found in many range images is speckle noise. This noise appears as perturbations of the real range values in most pixels. This noise poses a bigger problem than the previous one as attempts to remove it usually reduce the sharpness of the edges in the image, leading to erroneous results in subsequent segmentations. Several methods were experimented with for the elimination of this type of noise, and geometric filtering was found to yield minimal effect on edges in the image. The algorithm used is called the "eight hull algorithm" and was developed by Crimmins [14]. It uses a hull function that compares each pixel with its 8 surrounding neighbors to reduce the speckle without reducing the sharpness of the edges.

Both of these types of noise were encountered in the range data used in this work. The methods above were used successfully to reduce this noise. Figure 6.2 shows real range image TOOLS. When projected to 3–D coordinates, noise in the range data becomes visible, as is shown in Fig. 6.3. The result of application of the eight hull algorithm to the real range image TOOLS is shown at the bottom of Fig. 6.3. We notice that the edges are unaffected by this procedure, but speckle noise is significantly reduced.

After noise removal, correction for coordinates is applied. The scanner used to produce many of the images in this work is an Odetics laser scanner. This scanner measures angular distances from a point by using an equal angle increment sampling of the azimuth and elevation angles. These measurements are

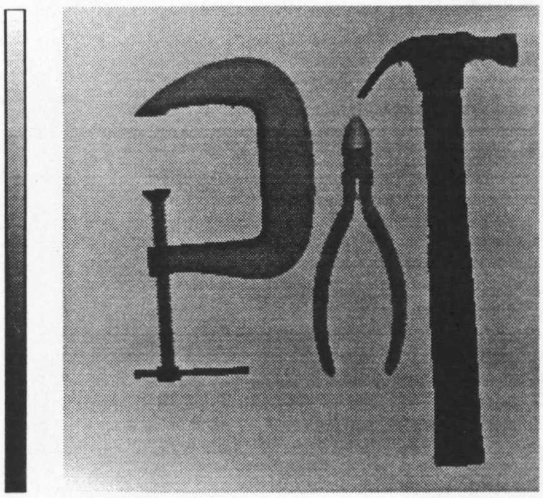


Figure 6.2: *Real range image TOOLS. The image was reduced from* 300 × 300 *to* 128 × 128 *pixels. (Original image provided by Coleman Research Corporation.)*

---

not perpendicular to the Z axis, and therefore need to be converted to rectangular Cartesian coordinates. Given an angular measurement (r, θ, ϕ), where r is the measured range, θ is the azimuth angle, and ϕ is the elevation angle, the Cartesian coordinates can be retrieved by [4]

$$x = \frac{r\tan(\theta)}{\sqrt{1+\tan^2(\theta)+\tan^2(\phi)}}$$

$$y = \frac{r\tan(\phi)}{\sqrt{1+\tan^2(\theta)+\tan^2(\phi)}}$$

$$z = \frac{r}{\sqrt{1+\tan^2(\theta)+\tan^2(\phi)}} \ .$$

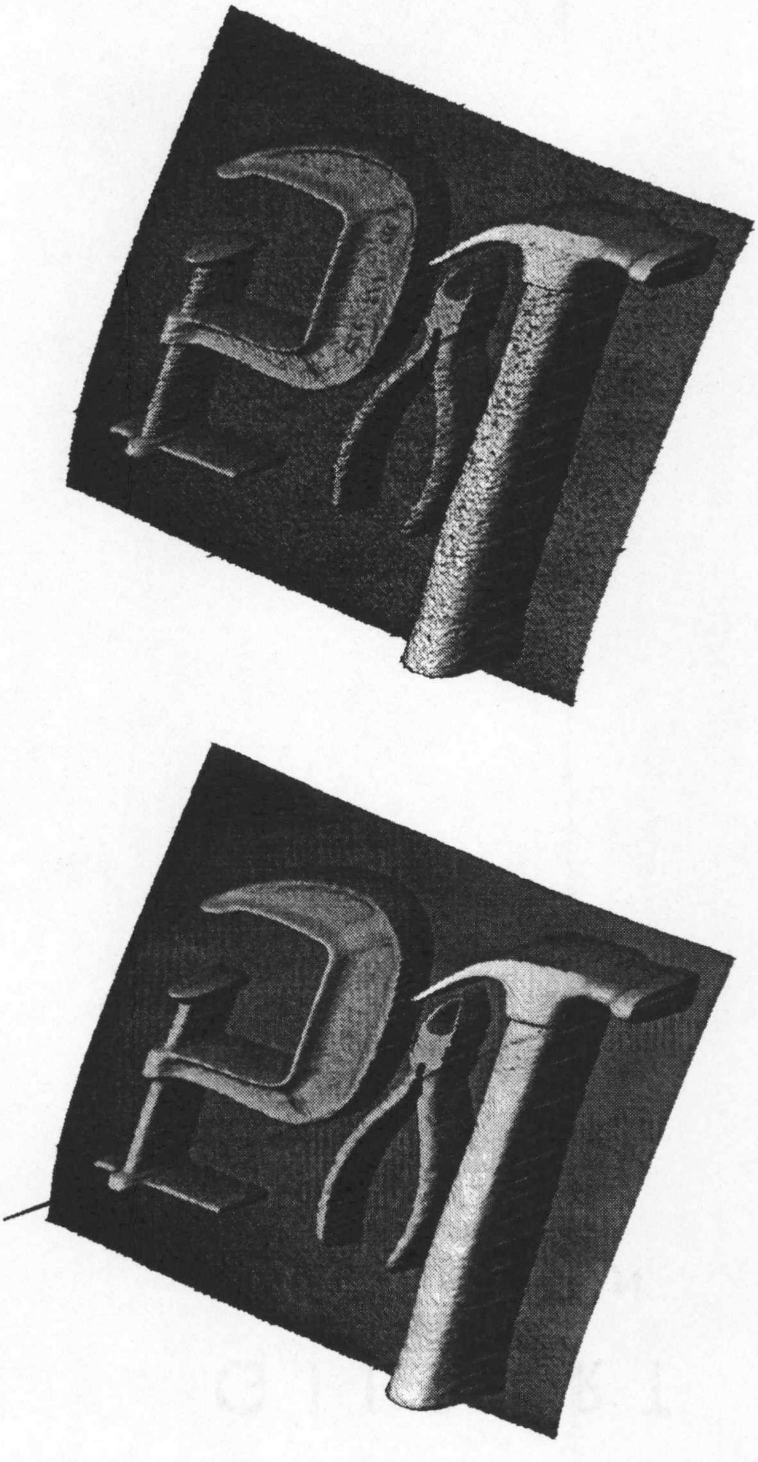


Figure 6.3: *Real range image TOOLS before (top) and after (bottom) geometric filtering.*

6.1.2 The *À Trous* Module

The second module in Fig. 6.1 is the *à trous* module that applies our implementation of the DWT as given in Chapter 3. This module uses two inputs, the preprocessed range image and the wavelet to be applied to it. The output of this module is estimates for derivatives in the vertical and horizontal directions, labeled $W^1 f(x,y)$ and $W^2 f(x,y)$, respectively.

6.1.3 Modulus Maxima Tracing

The third module shown in Fig. 6.1 calculates the modulus and angle of the transform from $W^1 f(x,y)$ and $W^2 f(x,y)$. Then, the modulus maxima are traced to generate a binary edge map. The modulus of the WT at the scale s is given by

$$M_s f(x,y) = \sqrt{|W_s^1 f(x,y)|^2 + |W_s^2 f(x,y)|^2}\ .$$

These values are scaled between 0 and 255. The angle of the WT at the scale s is given by

$$A_s f(x,y) = \tan^{-1}\left(\frac{W_s^2 f(x,y)}{W_s^1 f(x,y)}\right)\ .$$

The angle is calculated to give $-\pi \leq A_s f(x,y) < \pi$. This angle is then scaled between 0 and 255. To minimize spike noise, values of W_1 or W_2 falling in the lower and upper 5% are set equal to zero. This reduces the problem of dealing with arithmetic on very small floating numbers.

Once the modulus and the angle images have been calculated, we recover the wavelet maxima from the modulus image by finding maxima points in the direction of the angle. For each pixel m in the modulus image, we look up the corresponding pixel value in the angle image $a_{n,m}$ and get the direction of the gradient. If we let the pixel in the modulus image be m_5, we look up the value of the angle image a_5 at that location and we detect a maxima at m_5 in the following instances:

$$
\begin{array}{lll}
\text{If} & \frac{-\pi}{8} < a_5 \leq \frac{\pi}{8}, & m_4 \leq m_5 < m_6 \\
& \frac{+\pi}{8} < a_5 \leq \frac{+3\pi}{8}, & m_7 \leq m_5 < m_3 \\
& \frac{+3\pi}{8} < a_5 \leq \frac{+5\pi}{8}, & m_8 \leq m_5 < m_2 \\
& \frac{+5\pi}{8} < a_5 \leq \frac{+7\pi}{8}, & m_9 \leq m_5 < m_1 \\
& a_5 < \frac{-7\pi}{8} \text{ or } a_5 > \frac{7\pi}{8}, & m_6 \leq m_5 < m_4 \\
& \frac{-7\pi}{8} < a_5 \leq \frac{-5\pi}{8}, & m_3 \leq m_5 < m_7 \\
& \frac{-5\pi}{8} < a_5 \leq \frac{-3\pi}{8}, & m_1 \leq m_5 < m_9 \\
& \frac{-3\pi}{8} < a_5 \leq \frac{-1\pi}{8}, & m_2 \leq m_5 < m_8
\end{array}
$$

The calculations are shown in Fig. 6.4. This is repeated for the whole of the modulus image, and a binary maxima image is generated, indicating at each point whether or not a maxima exists at that location.

6.1.4 Experimental Results

Figure 6.5 shows the results of applying the modulus maxima algorithm to image TOOLS. This image is not rich in surface features but is used here to illustrate the steps involved. All the images shown are at the first scale. The image $W^1f(x,y)$ contains the vertical discontinuities or edges, calculated as we make a horizontal pass on the image. The $W^1f(x,y)$, on the other hand, shows the horizontal edges. The modulus image, $Mf(x,y)$, combines these two surface discontinuity

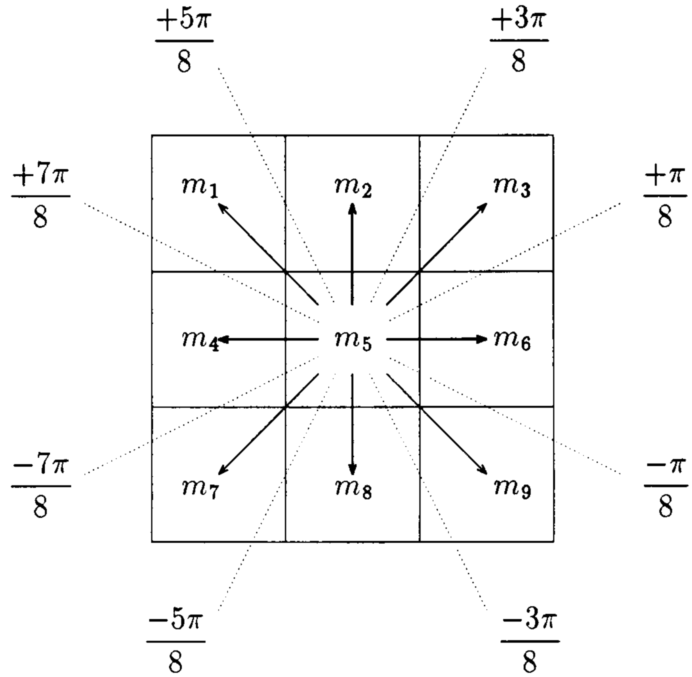


Figure 6.4: *Tracing of wavelet maxima using angle image.*

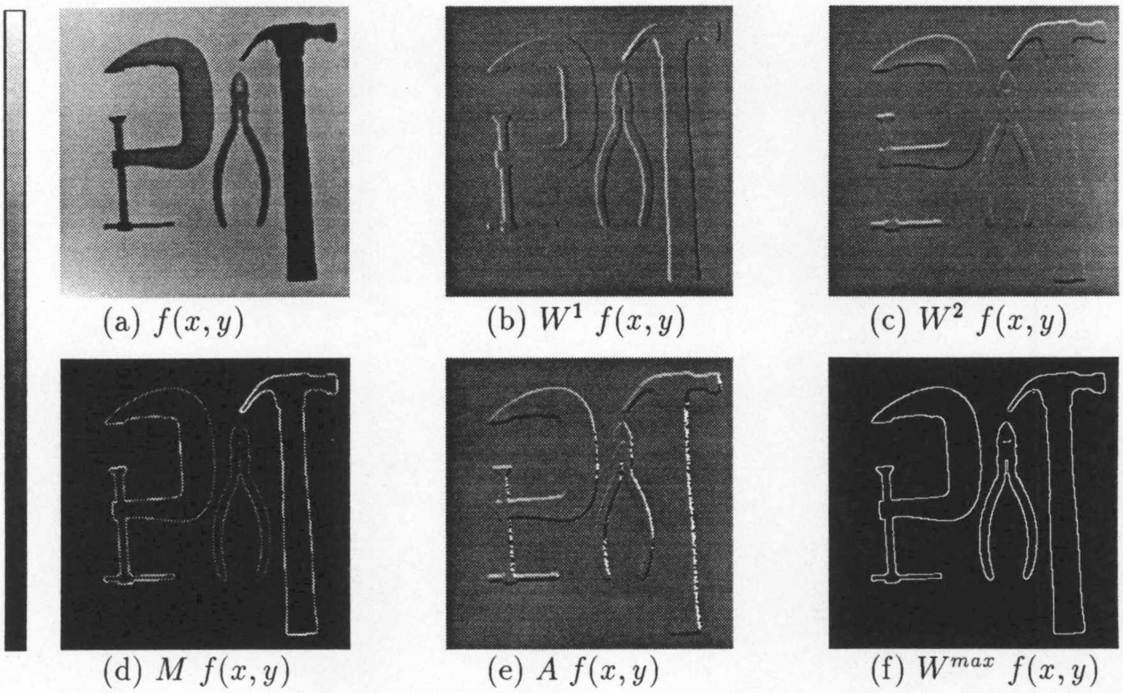
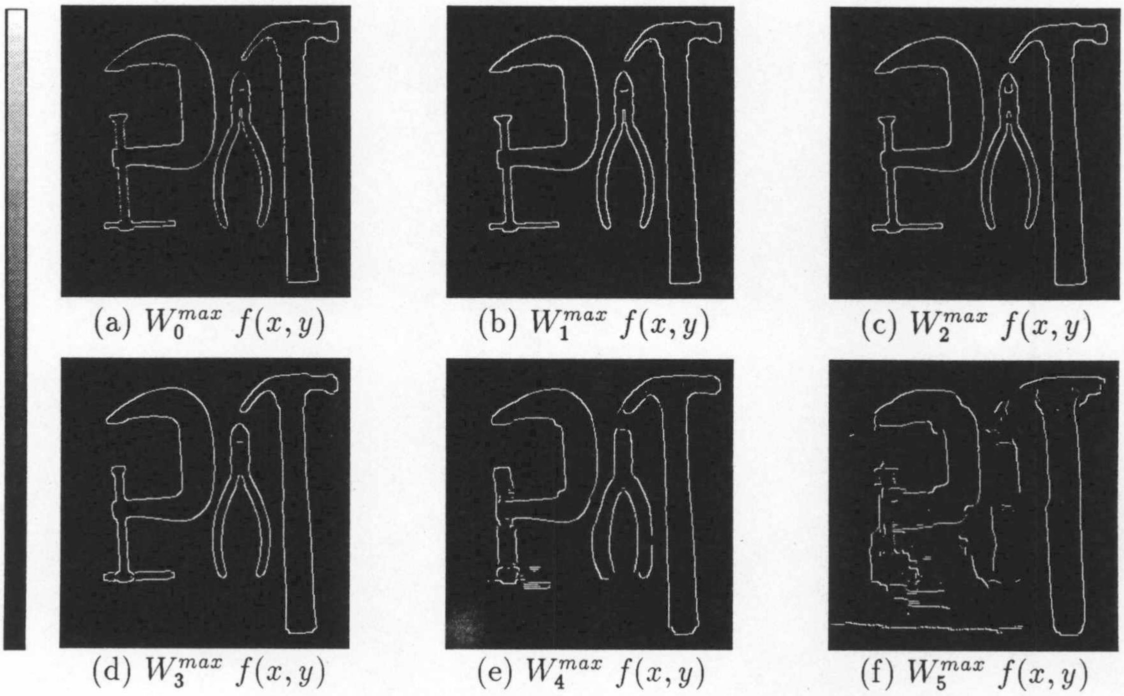


Figure 6.5: *Segmentation of real range image TOOLS using wavelet maxima.*

maps into one image, while the angle image $Af(x,y)$ gives the direction in which the sharpest variation occurs. Finally, the modulus maxima image, $W^{max}f(x,y)$ shows the edges obtained by tracing the maxima in $Mf(x,y)$ at the angle given by $Af(x,y)$. We note that all the prominent edges in the original image are represented.

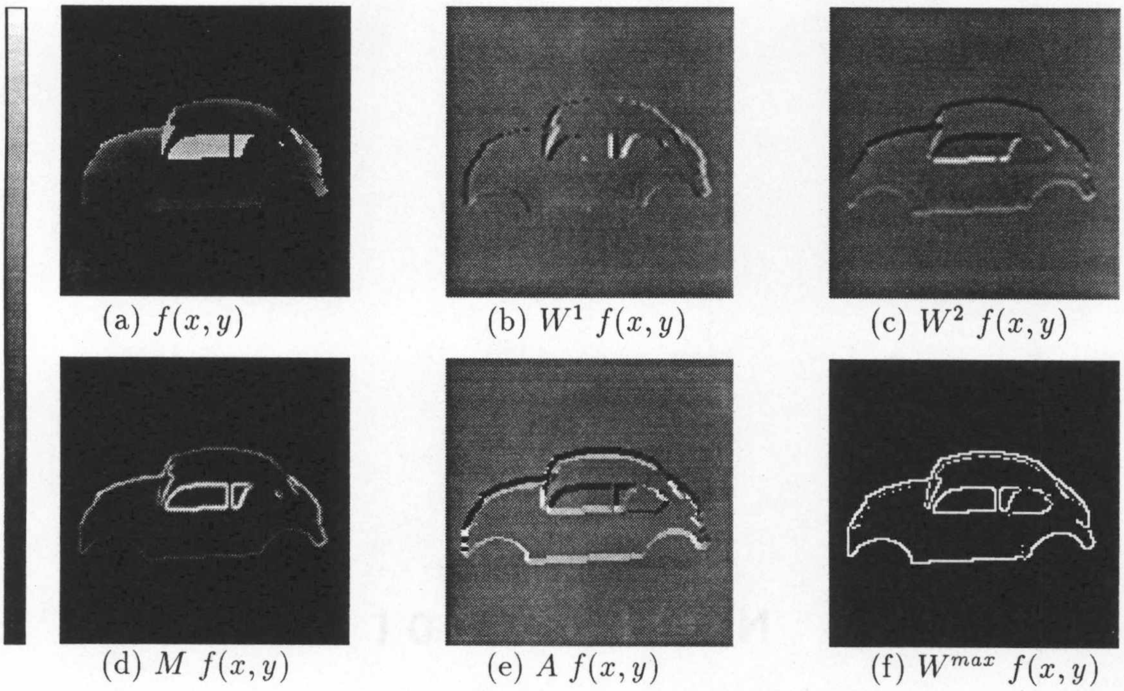
Figures 6.6–6.10 show the application of this method to several other images of increasing complexity. All the results are shown at the first scale, which was experimentally found to provide enough smoothing without reducing the sharpness of the edges and surfaces in the images.



(a) $W_0^{max}\ f(x,y)$ (b) $W_1^{max}\ f(x,y)$ (c) $W_2^{max}\ f(x,y)$

(d) $W_3^{max}\ f(x,y)$ (e) $W_4^{max}\ f(x,y)$ (f) $W_5^{max}\ f(x,y)$

Figure 6.6: *Final edge map for image TOOLS at scales* 0 *to* 5.



(a) $f(x,y)$ (b) $W^1\ f(x,y)$ (c) $W^2\ f(x,y)$

(d) $M\ f(x,y)$ (e) $A\ f(x,y)$ (f) $W^{max}\ f(x,y)$

Figure 6.7: *Segmentation of synthetic range image VW using wavelet maxima.*

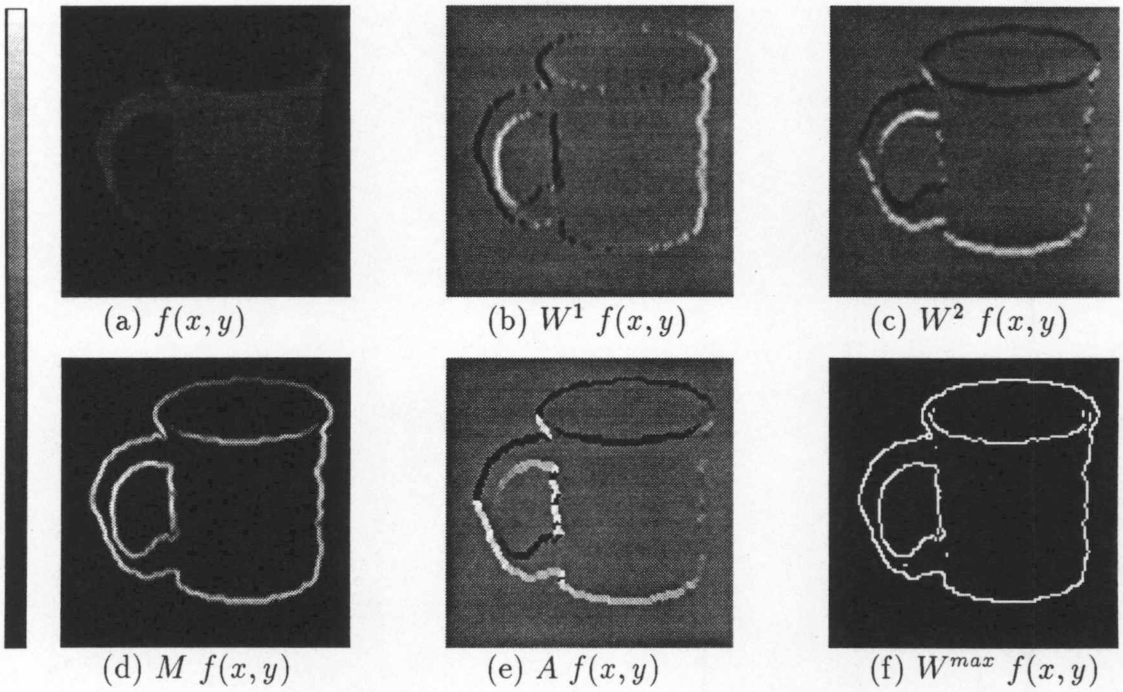


Figure 6.8: *Segmentation of real range image CUP using wavelet maxima.*

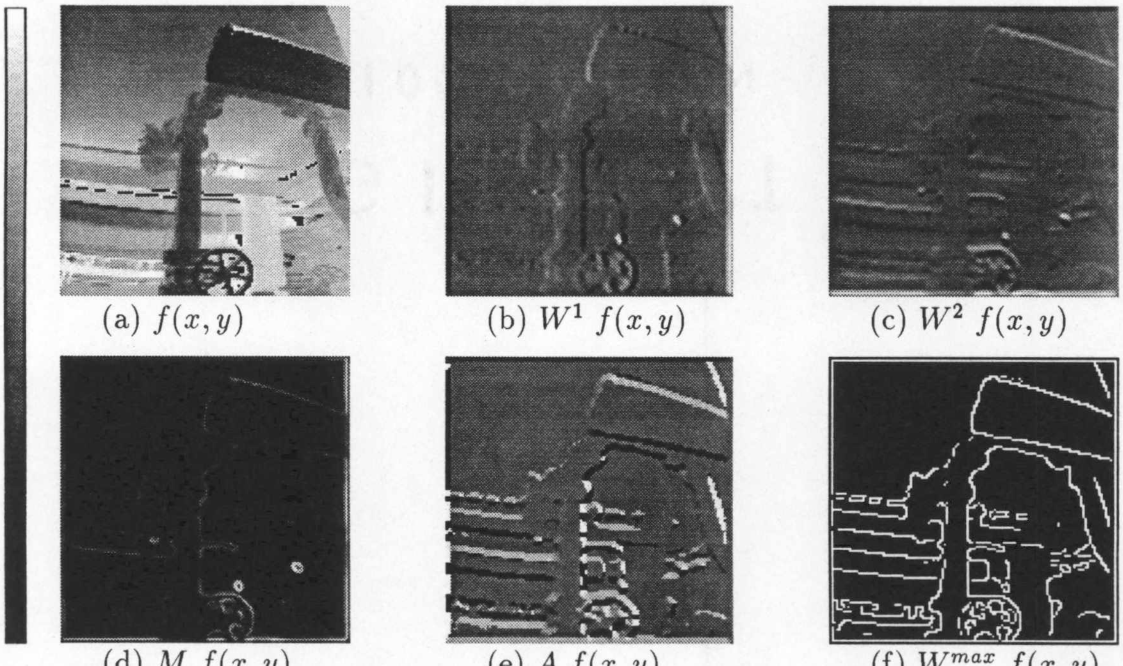
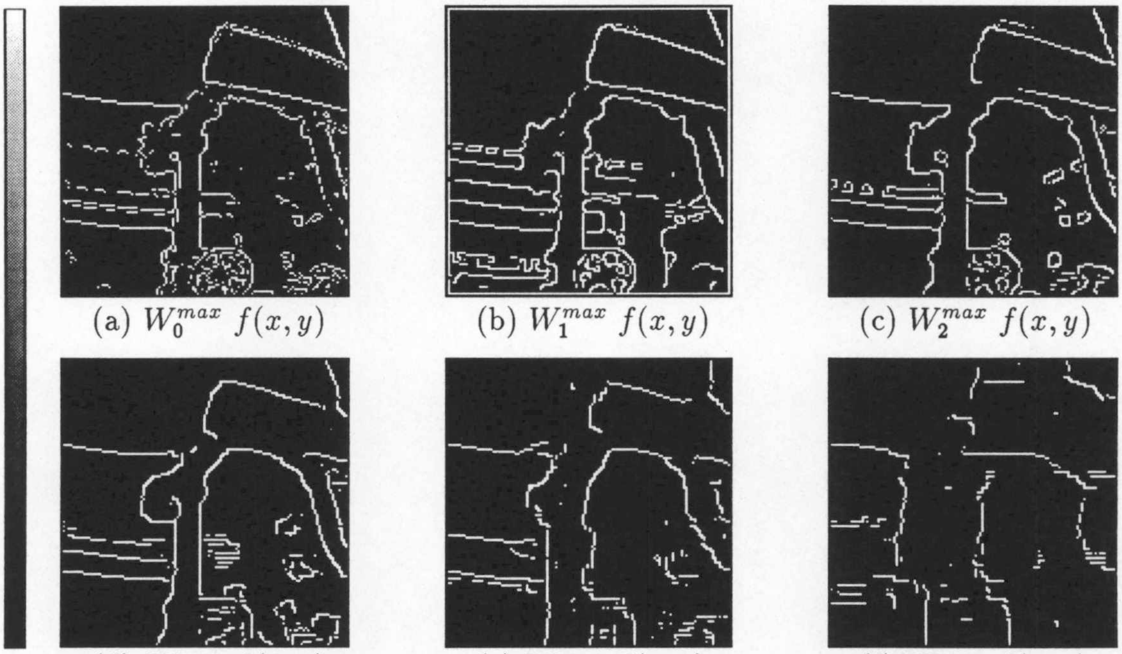


Figure 6.9: *Segmentation of real range image PIPES using wavelet maxima.*



(a) $W_0^{max} f(x,y)$ (b) $W_1^{max} f(x,y)$ (c) $W_2^{max} f(x,y)$

(d) $W_3^{max} f(x,y)$ (e) $W_4^{max} f(x,y)$ (f) $W_5^{max} f(x,y)$

Figure 6.10: *Final edge map for image PIPES at scales* 0 *to* 5.

6.1.5 Analysis of Results

The segmentation result for image TOOLS in Fig. 6.5 at scale 1 shows that all the step edges at the boundary of the objects are detected. Further, very little noise exists in the final edge map. The edge detection process was applied both to the original image and to the filtered image, yielding approximately the same result. Since the modulus maxima has smoothing built into the edge detection, the speckle noise apparent in Fig. 6.3 does not affect the outcome of the algorithm.

Figure 6.6 shows the modulus maxima for image TOOLS at the scales from 0 to 5. As would be expected, scale 0 is often susceptible to noise because it does not provide any smoothing. As the scale increases, the modulus maxima image increasingly focuses on the macro features of the image. Eventually, interedge interactions become apparent, introducing distorted contours due to the length of the filter (19 coefficients at scale 5). We notice that there is very little difference between the scales 1, 2, and 3. The filter coefficients at these scales are 7, 10, and 13, respectively. For this particular image, the increase in scale (and hence smoothing) does not affect the outcome, as the original image contained features of roughly the same size.

The results for the images VW and CUP in Figs 6.7 and 6.8 demonstrate the ability of the modulus maxima algorithm to accurately trace curved boundaries. We note that the image $W^1\, f(x,y)$, which represents the vertical edges, also picks up other edges in the scene. This is also true for $W^2\, f(x,y)$. Thus, most edges are present in both images, but their strength varies inversely in both images. The images at 45° are represented equally in both images. This is particularly apparent in the segmentation of the image CUP.

For the image VW, we note that the algorithm detects the lower edge of the rear window. This edge is not apparent in the original image but does in fact exist, as was discovered by inspecting the pixel values.

The image PIPES in Fig. 6.9 represents a busy scene with features of various sizes. This image is used to illustrate the role of scale in the analysis. Figure 6.10 shows the modulus maxima at the scales 0 to 5. As we found with image PIPES, the modulus maxima at scale 0 is noisy due to the absence of smoothing. Scales 1 and 2 show the best results as they provide enough smoothing without noticeable interedge interference. As we move to higher scales, the edge contours become distorted due to the length of the filter, causing edges to interact with each other.

6.2 Wavelet Transform Angle Algorithm

This algorithm generates a complete segmentation of the input range image. The final output is labeled such that each surface in the scene is mapped to a specific intensity value. A block diagram of the main steps in this algorithm is shown in Fig 6.11. As before, a step edge map $W^{max}f(x,y)$ is generated from modulus maxima tracing acting directly on the input range image $f(x,y)$. However, this algorithm also uses the angle image $A\ f(x,y)$ as input to a second *à trous* module to find the horizontal and vertical components, $W^1(A\,f(x,y))$ and $W^2(A\,f(x,y))$, respectively. These are then used to calculate a modulus image $M(A\,f(x,y))$ and subsequently apply a second tracing of maxima on this modulus $M(A\,f(x,y))$. The result, $W^{max}(A\,f(x,y))$, is used to complement the edge map found earlier by fusing the information from $W^{max}f(x,y)$ and $W^{max}(A\,f(x,y))$. This gives a preliminary surface edge map.

The first step applied to this edge map is noise attentuation. This is done by checking for isolated sparse noise pixels. Since this is a binary image, the operation can be done without sacrifice of processing time.

The next step applied is edge linking. Most edge detection algorithms miss one or two pixels within edge contours. These gaps, if left, would lead to under–segmentation, since the labeling procedure would label both as the same region. The edge linking is done using morphological opening and closing. At first, we apply erosion to the binary and then we apply conditional dilation. We can control the dilation process to connect gaps up to any given spacing, such as 3 or 5 pixels.

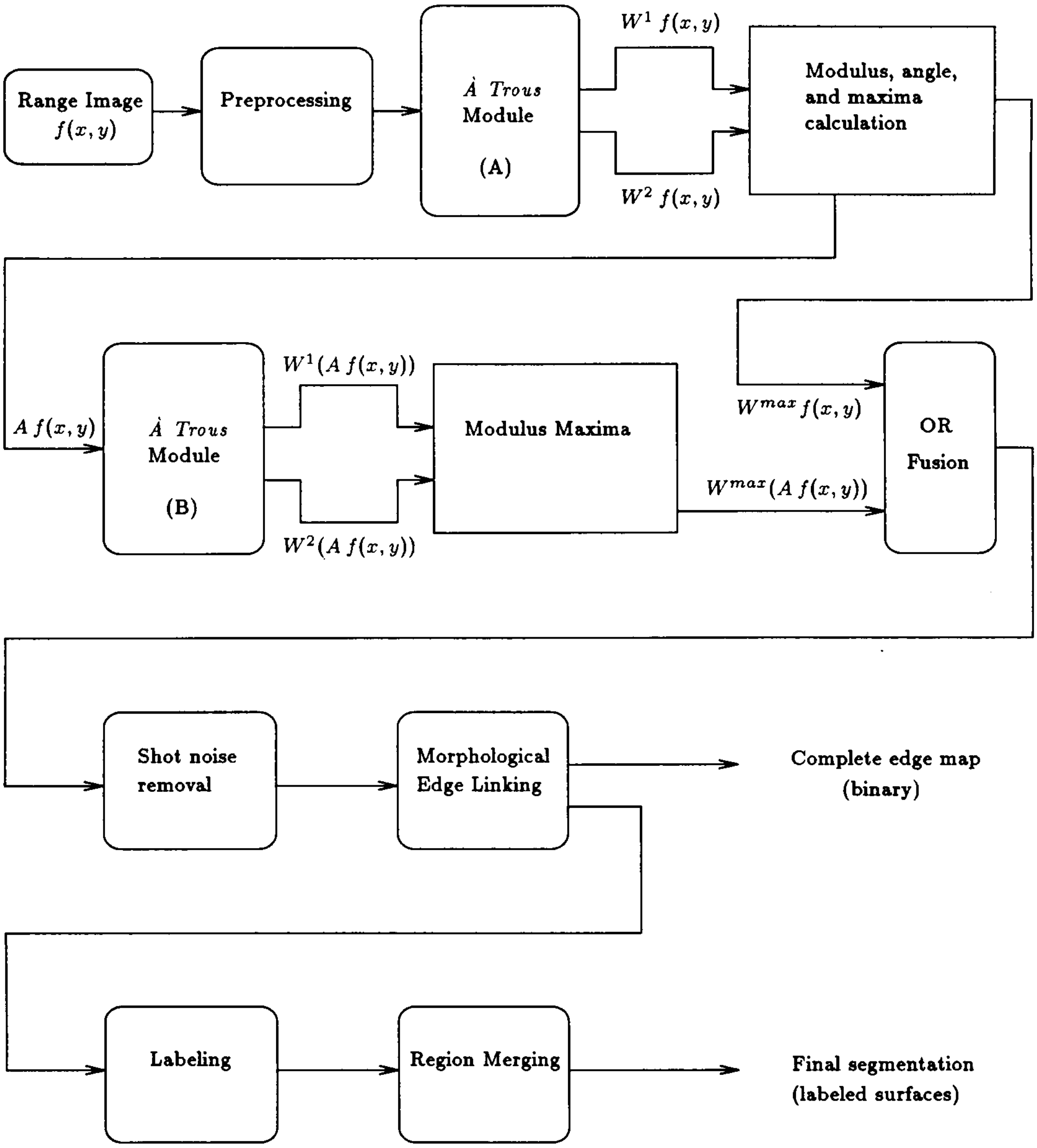


Figure 6.11: *Block diagram for modulus maxima of angle algorithm.*

The gap width has to be an odd number. Closed contours for most edges that allow for labeling are usually obtained with three pixels linking.

The output of the edge linking module is an image ready for labeling. In this work, labeling was done based on 4–connectivity [26] because 8–connectivity labeling tends to yield multiple path connections, giving rise to ambiguity in the labeling process. Four–connectivity labeling was experimentally found to yield acceptable results in every case and was thus adhered to throughout.

Region merging is applied to a labeled image to help avoid over–segmentation. Often, immediately after labeling, the image has many regions that can be merged or eliminated altogether. These small regions appear as speckles in the segmented image. Region merging is achieved by inspecting the histogram of the labeled image and finding regions that have a small number of pixels, say less than 15. We assume that these regions are too small to be distinct surfaces in the scene and attempt to merge them with the rest of the image. If this small region is wholly contained within another region, we can relabel the smaller region by inspecting its connectivity. If the small region is adjacent to more than one other region, it is more difficult to attempt to merge it and we simply set to it to 0. In the latter case, it still appears in the final image but can be distinguished from the rest of the image as an indeterminant region.

6.2.1 Analysis of Results

Results for this algorithm are shown in Figs. 6.12–6.17. Figure 6.13 shows the steps involved in the segmentation of the synthetic range image CURVBLOCK. The

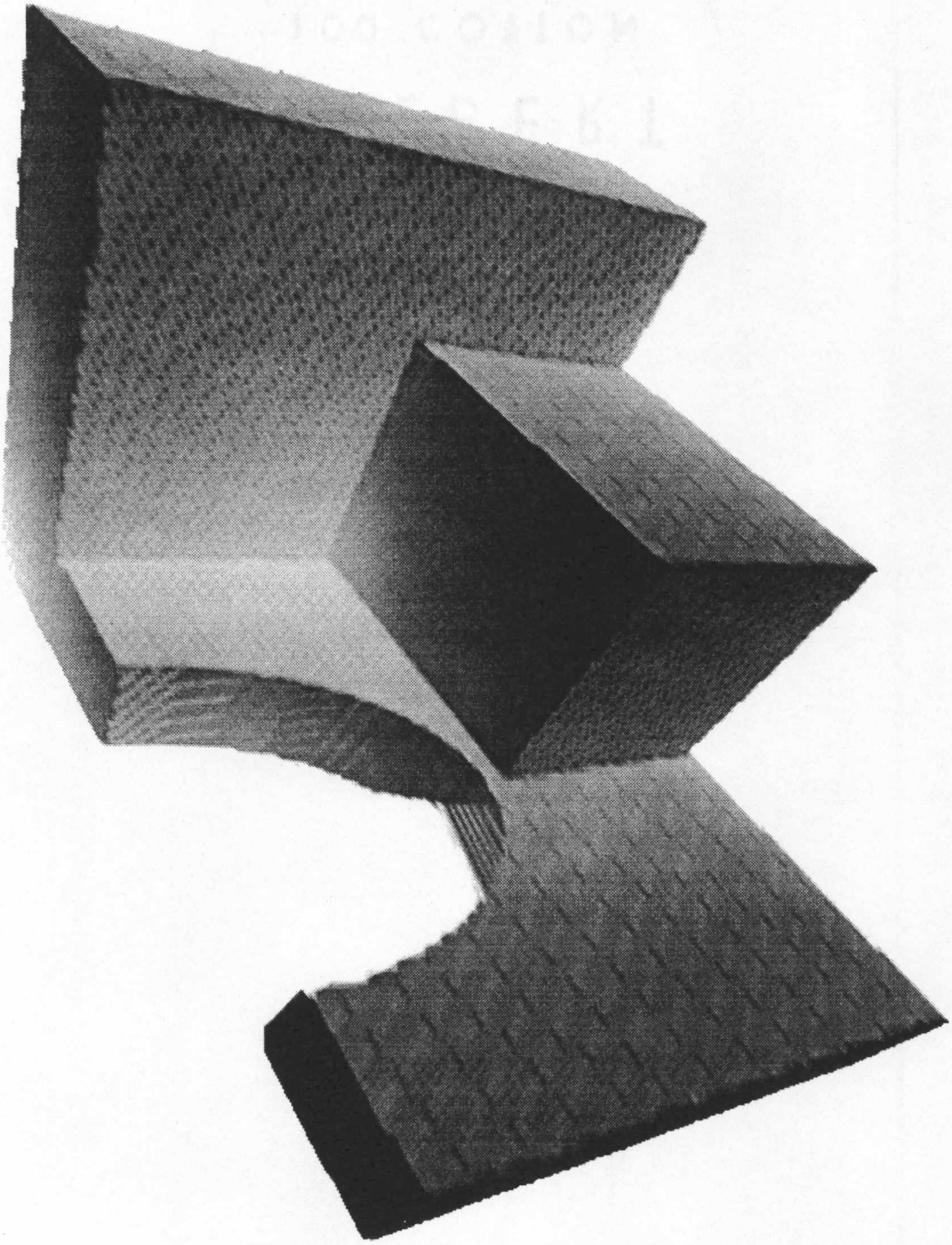
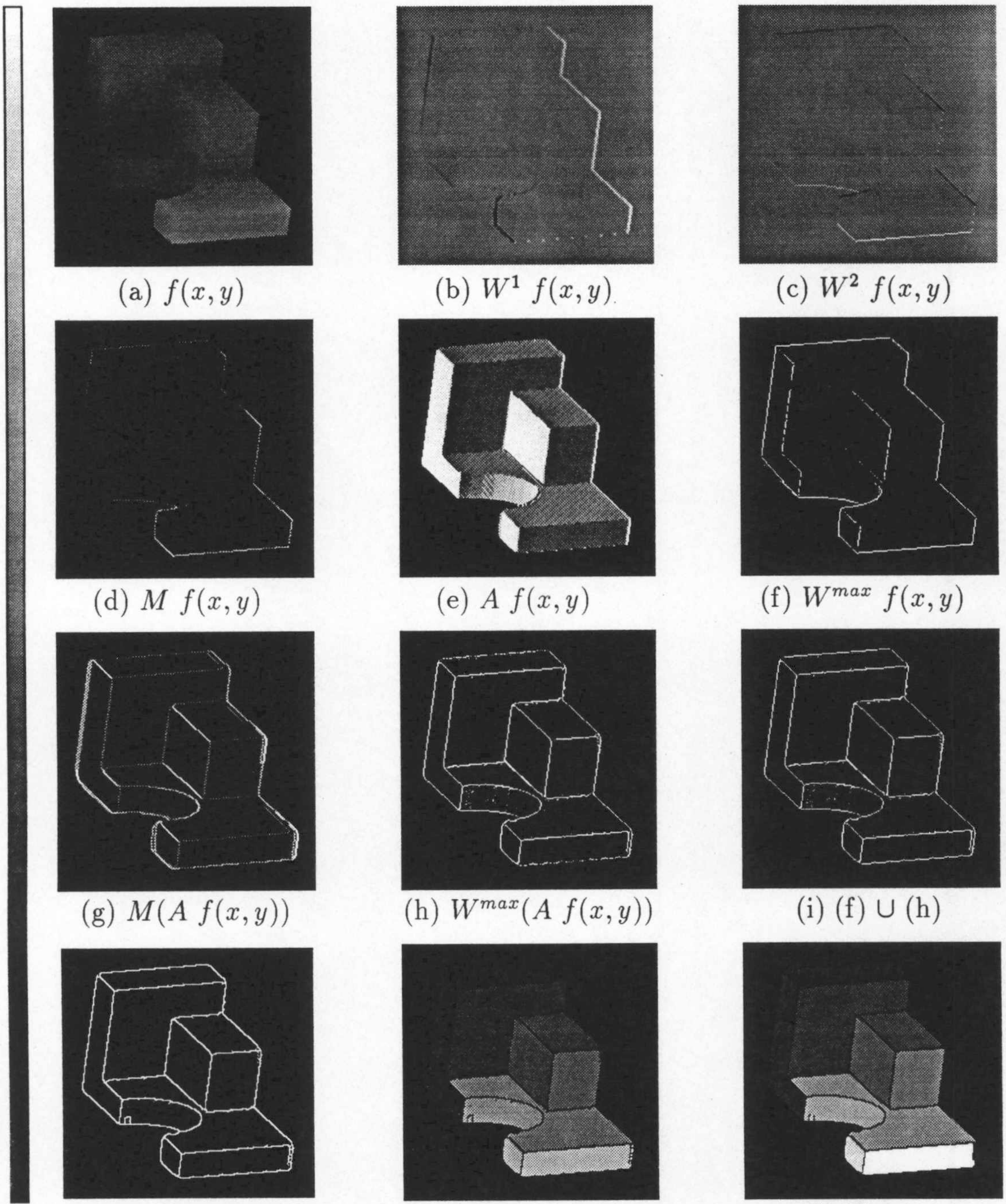


Figure 6.12: *A 3–D projection of synthetic range image CURVBLOCK. (Original range image obtained from the Michigan State/Washington State Universities Range Image Database.)*



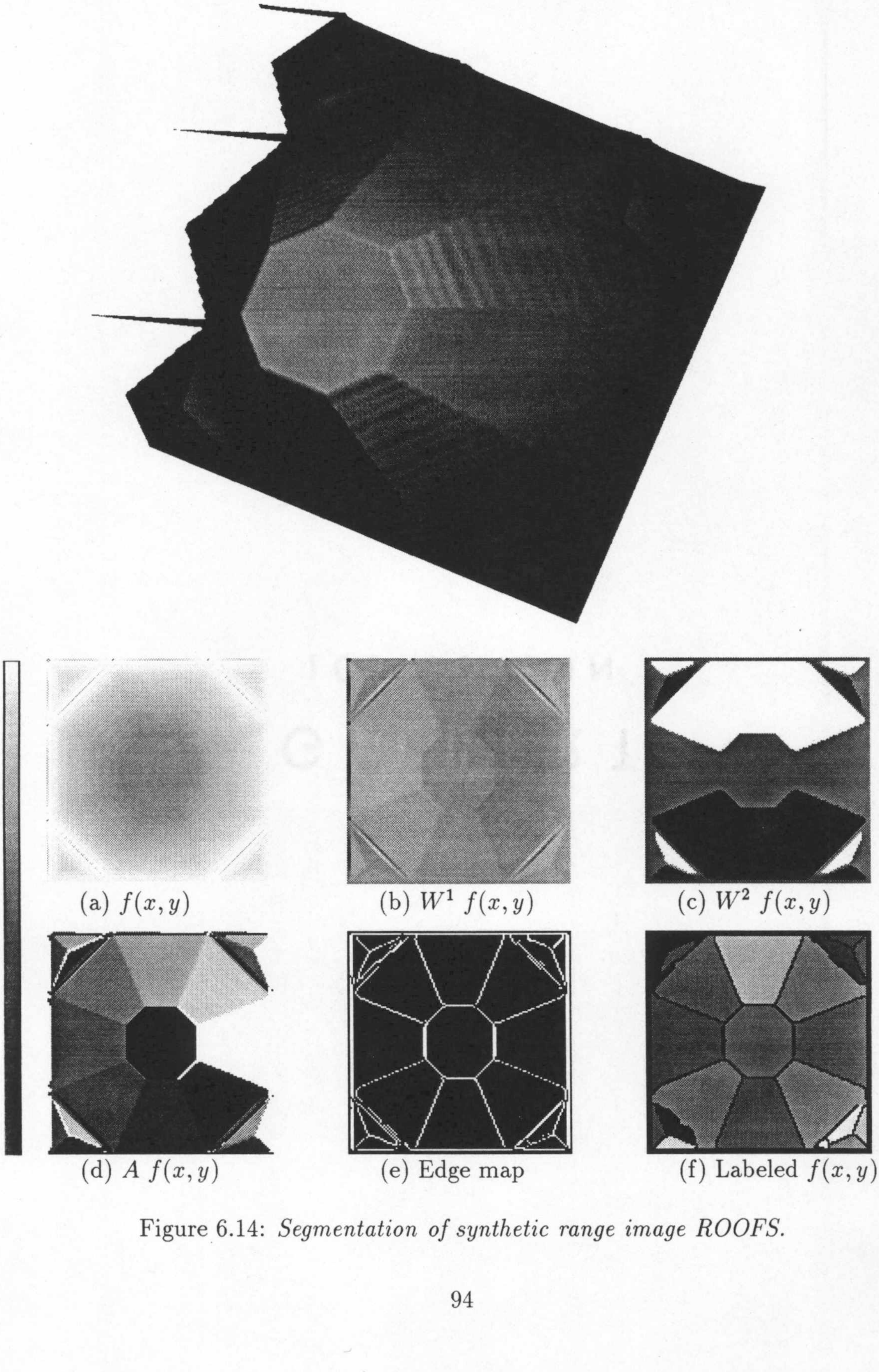
(a) $f(x,y)$ (b) $W^1\ f(x,y)$ (c) $W^2\ f(x,y)$

(d) $M\ f(x,y)$ (e) $A\ f(x,y)$ (f) $W^{max}\ f(x,y)$

(g) $M(A\ f(x,y))$ (h) $W^{max}(A\ f(x,y))$ (i) (f) $\cup$ (h)

(j) Processing of (i) (k) Initial labeling (l) Final

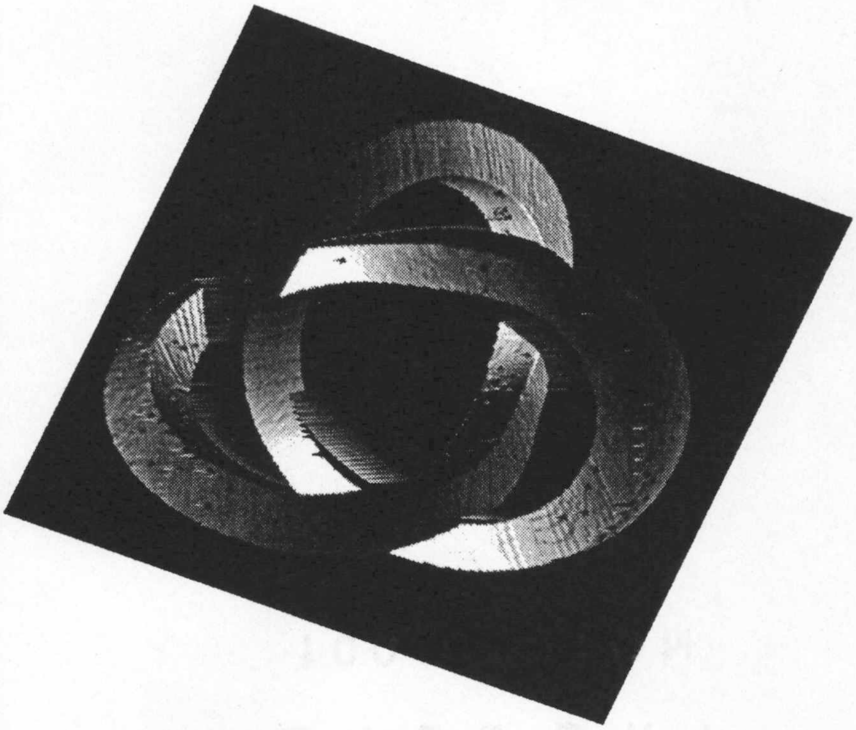
Figure 6.13: *Segmentation of image CURVBLOCK showing intermediate results.*

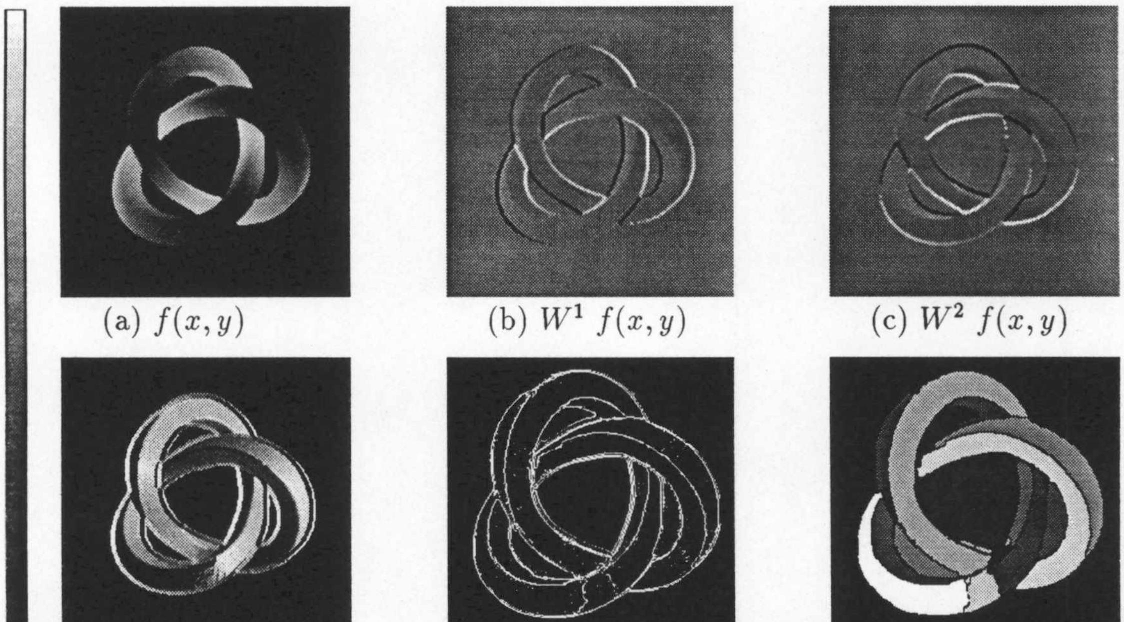


(a) $f(x,y)$ (b) $W^1\ f(x,y)$ (c) $W^2\ f(x,y)$

(d) $A\ f(x,y)$ (e) Edge map (f) Labeled $f(x,y)$

Figure 6.14: *Segmentation of synthetic range image ROOFS.*

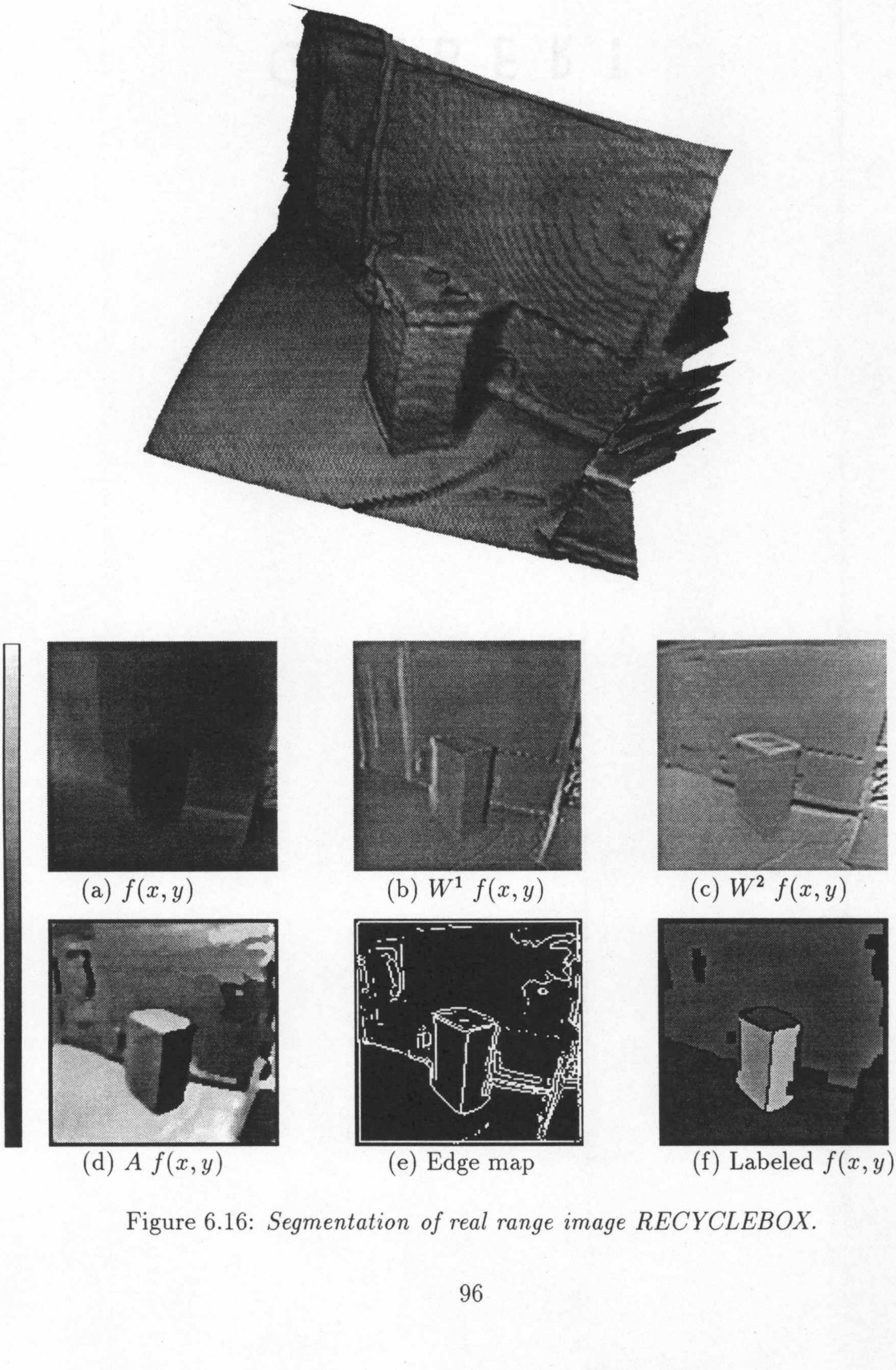




(a) $f(x, y)$ (b) $W^1\ f(x, y)$ (c) $W^2\ f(x, y)$

(d) $A\ f(x, y)$ (e) Edge map (f) Labeled $f(x, y)$

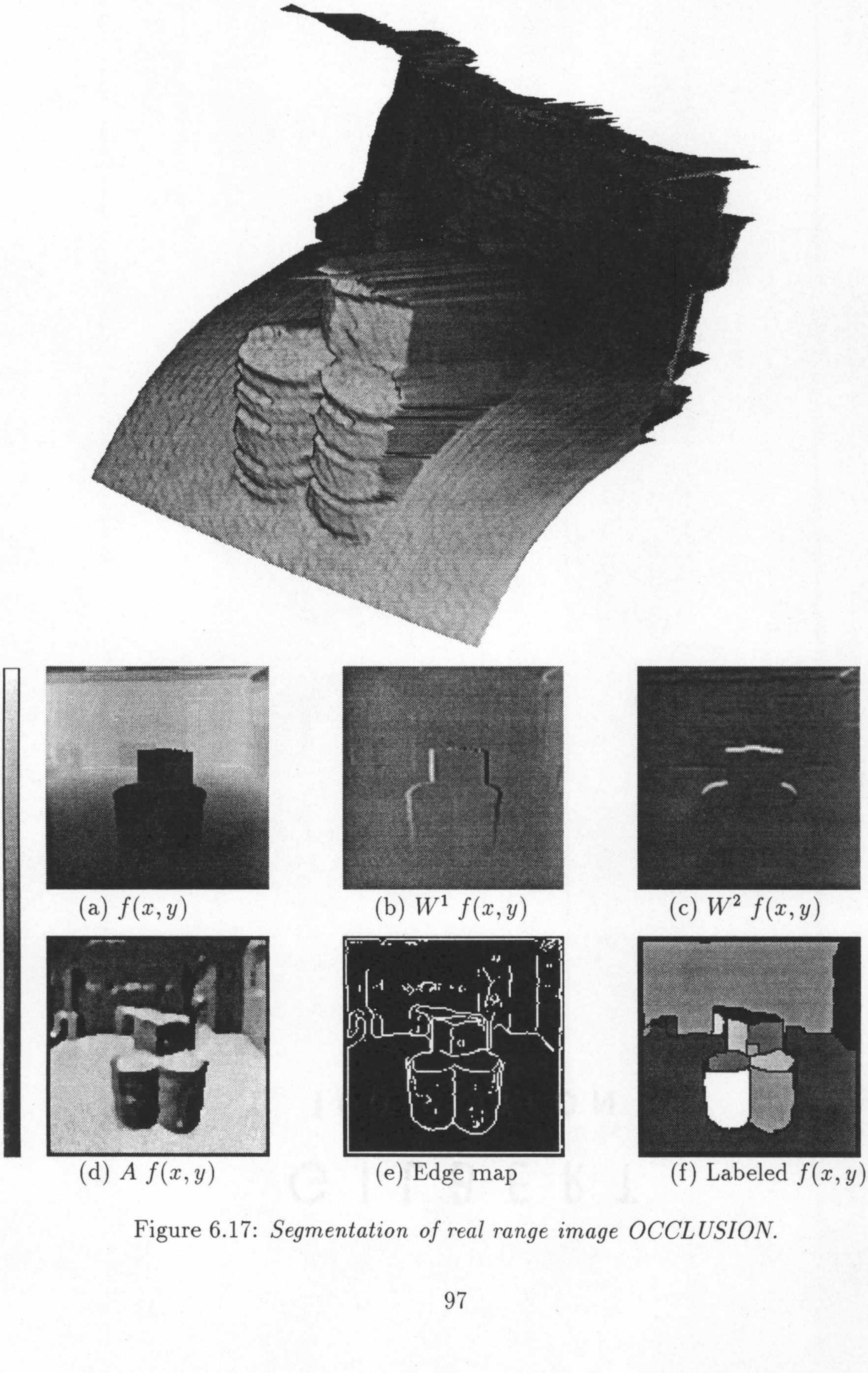
Figure 6.15: *Segmentation of synthetic range image ESCHERKNOT.*



(a) $f(x,y)$ (b) $W^1\ f(x,y)$ (c) $W^2\ f(x,y)$

(d) $A\ f(x,y)$ (e) Edge map (f) Labeled $f(x,y)$

Figure 6.16: *Segmentation of real range image RECYCLEBOX.*



(a) $f(x,y)$ (b) $W^1\ f(x,y)$ (c) $W^2\ f(x,y)$

(d) $A\ f(x,y)$ (e) Edge map (f) Labeled $f(x,y)$

Figure 6.17: *Segmentation of real range image OCCLUSION.*

original image is shown in Fig. 6.13(a). The vertical and horizontal components are shown in Fig. 6.13(b) and Fig. 6.13(c), respectively. We note that the curved insert into the block appears in both Fig. 6.13(b) and Fig. 6.13(c). Image Fig. 6.13(d) shows the magnitude of the WT. All the step edges in the original image appear in the magnitude image. The angle image Fig. 6.13(e) tells us the direction of the steepest descent for each surface or plane in the image. Each surface facing a direction is assigned the value of the angle at which the vertical plane that the surface is tilted. Thus, all the sides of the block that point to the top of it have the same value. The image Fig. 6.13(f) gives the modulus maxima of the magnitude image Fig. 6.13(d) and the angle image Fig. 6.13(e). This image indicates the step edges in the image.

The angle image Fig. 6.13(e), which contains a mapping of the surface orientations, is now used as an input to the *à trous* module to give a magnitude of angle image Fig. 6.13(g). This image contains the edges between surfaces of different orientations as calculated by the angle image Fig. 6.13(e). These edges appear as roof edges in the original image. The edges detected in the image Fig. 6.13(h) are now combined with the earlier edge map Fig. 6.13(f) to give a complete binary edge map of the scene. This edge map is first processed for binary noise, and then edge linking is applied. The edge map in Fig. 6.13(j) is now ready for labeling, as all the contours have been closed by the edge linking procedure. The image Fig. 6.13(k) shows the initial labeling of the processed edge map. This initial labeling contains very small regions that appear as noise. These are then merged with larger patches. Image Fig. 6.13(l) shows the final result.

The image ROOFS in Fig. 6.14 shows a synthetic range image consisting only of roof edges at different orientations. The vertical components image Fig. 6.13(b) contains all the surface discontinuities except the perfectly horizontal ones. Similarly, the horizontal components image Fig. 6.13(c) contains all but the vertical surface discontinuities. The angle image Fig. 6.13(d) distinguishes the orientations within the image by assigning a different value for each surface. The discontinuities appear in the final result as edges.

The image ESCHERKNOT in Fig. 6.15 shows a synthetic range image consisting of three curved strips interwined to form a knot. This image is used to illustrate how the algorithm copes with curved surfaces. The angle image Fig. 6.15(d) successfully traces the angle of the surfaces, showing a smooth transition along each of the stripes. We note that a rupture occurs at the bottom of the stripe in the front. This happens as the angle of the surface reaches its maximum value and wraps around the range $-\pi$ to π. This rupture also appears in the final result Fig. 6.15(f).

The image RECYCLEBOX in Fig. 6.16 shows a real range image with a recycling box for aluminum cans in the middle. The 3–D projection of the image shows several defects in the image, particularly the elliptical coordinate effect that appears in the background. Also, a wire on the floor appears much exaggerated due to its different color and reflectivity properties. The angle image Fig. 6.16(d) shows several discontinuities in the background of the image. However, most of these discontinuities disappear in the final result as they do not form closed contours and the post–labeling region merging eliminates these patches. Some black patches still appear in the final result. These are regions with a pixel count higher than 15.

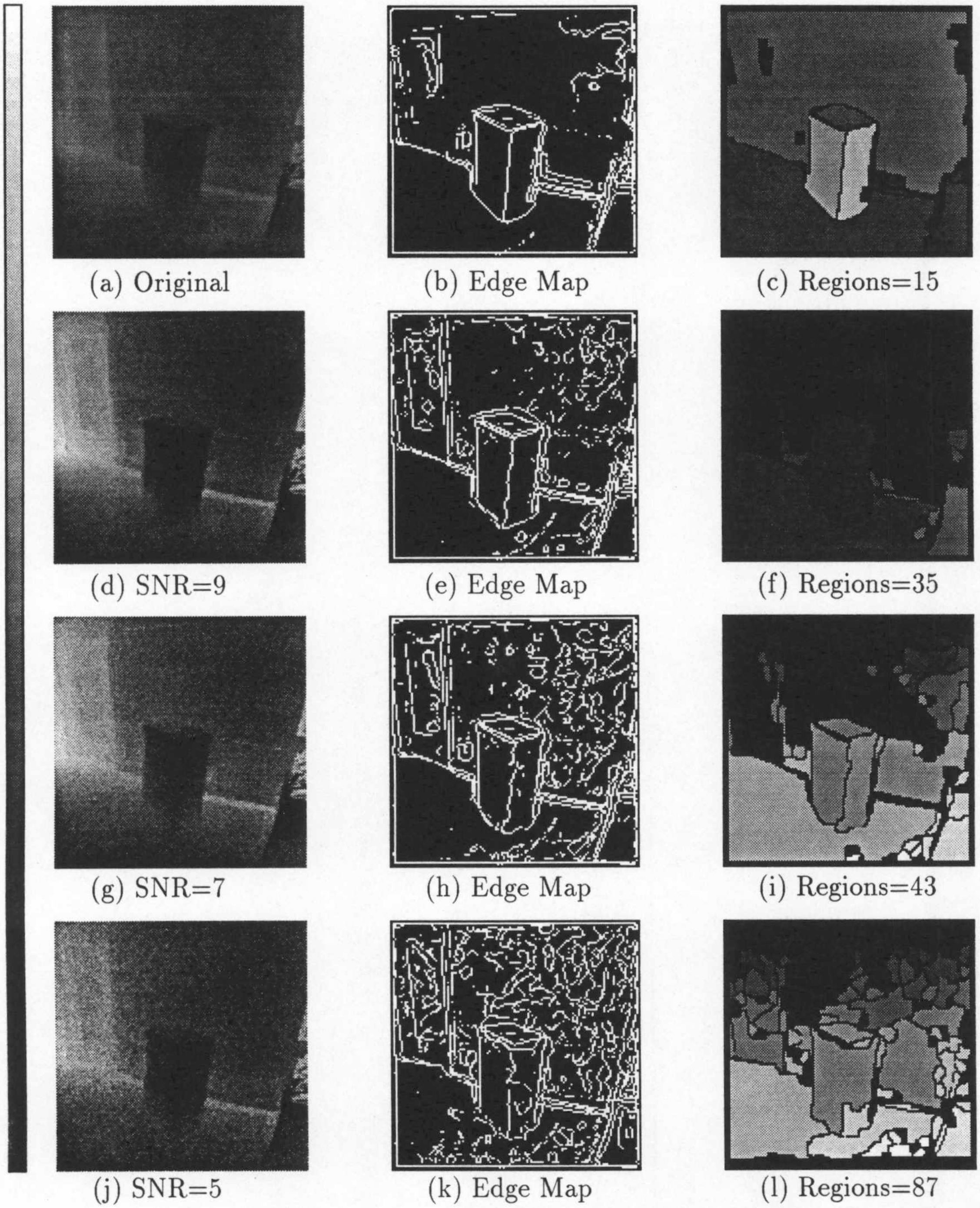
The image OCCLUSION in Fig. 6.17 shows a real range image of two barrels occluding a box. The 3–D projection of the image shows several irregular areas in the background of the image. The segmentation of this image illustrates the ability of this algorithm to eliminate several noisy areas from the background without sacrificing the level of detail in the final result. Further, the algorithm is able to discern several small regions such as the top of the box and the barrels.

6.3 Noise Performance of the Wavelet Transform Angle Algorithm

In this section we present results for the segmentation algorithm to test its robustness to noise. For the purpose of these experiments, an implementation of the Box–Muller algorithm [7] for generating random deviates with a Gaussian distribution was used. This algorithm was used to generate Gaussian noise to be added to the test image.

The real image used to test the noise performance is RECYCLEBOX. This image was chosen because the original laser–scanned image had minimal noise compared with the others. This can be seen by comparing its 3–D projection in Fig. 6.16 with projections of the other real images used in this work.

Figure 6.18 shows the segmentation results for the test image in the p-resence of zero–mean Gaussian noise with increasing standard deviation for the noise. The figure shows results for the original uncontaminated image as



(a) Original (b) Edge Map (c) Regions=15

(d) SNR=9 (e) Edge Map (f) Regions=35

(g) SNR=7 (h) Edge Map (i) Regions=43

(j) SNR=5 (k) Edge Map (l) Regions=87

Figure 6.18: *Effect of noise on segmentation of real range image RECYCLEBOX, showing the images used and the number of regions in the final segmentation.*

well as for $\sigma_n = \{3.26, 5.16, 8.19\}$. Using the definition signal–to–noise (SNR) $= 10\log(\sigma_i/\sigma_n)$ where σ_i is the standard deviation of the test image, the results shown correspond to SNR= $\{10, 9, 7, 5\}$. These results were obtained by contaminating the test image with noise immediately before the first application of the *à trous* algorithm. This was done in order to test the algorithm's robustness to noise, without regard to the performance of the prepocessing module that would normally attenuate some of the noise in the image.

Analysis of the segmentation results indicates that the noise added leads to over–segmentation of the image, yielding an increasing number of regions in the final segmented range. This is particularly evident in the background of the scene. However, the box and the edges associated with it are detected at all levels of noise used.

6.4 Time Analysis for the Wavelet Transform Angle Algorithm

This section investigates the computational complexity and the time performance of the WT angle algorithm for the segmentation of range images.

The DWT used in this work was implemented directly as a filter bank. The computations can be decomposed into convolutions and the subsampling operations or decimations that occur at each stage. The wavelets used are implemented as two filters of equal length. The decimation at each stage i was replaced by inserting $2^i - 1$ zeros between each two coefficients of the filters. The convolution was implemented so as to avoid these zeros, and therefore takes a fixed amount of time to calculate at each stage. Assuming a filter length of L, we need

$$L \text{ multiplications/pixel/stage and } L-1 \text{ additions/pixel/stage.}$$

At each stage, two images at the scale s are calculated, $W_s^1 f(x,y)$ and $W_s^2 f(x,y)$. Assuming the original image to be $N \times N$, each of these two images would have the same size. Thus the total number of operations needed to generate these two images is

$$2LN^2 \text{ multiplications and } 2(L-1)N^2 \text{ additions.}$$

It should be noted that more efficient implementations of the DWT exist [48]. These implementations apply known fast convolution techniques, such as the FFT, in a suitable manner. The computational complexity can then be reduced from L to $\log L$, which is significant for large filter lengths L, and the overall complexity for an image of dimension N can be reduced from $O(N^2)$ to $O(N \log N)$ complexity.

The calculation of the modulus of the WT image use two multiplications, one addition and one square root operation per pixel. The angle image is calculated using one inverse trigonometric function (tangent) per pixel. Analysis of the calculation of the wavelet maxima from the modulus and the angle images is more difficult as it contains several mathematical operations as well as an indeterminant number of comparisons that can range from 1 to 7 per pixel.

Actual times needed for the calculation of a wavelet maxima image from a given input image for different sizes for the input image are given in Table 6.1.

The figures given were obtained on a Sun SPARCstation 10. The times taken by the operation were calculated using the UNIX Operating System execution

Table 6.1: Time taken for calculation of wavelet maxima image.

Image Size	Time(sec)
32×32	0.09
64×64	0.25
128×128	1.08
256×256	3.97
512×512	15.58

profiling utility *prof*. The execution times obtained confirm the $O(N^2)$ complexity of the algorithm where N is the dimension of the image.

Total times for a complete segmentation of a range image are shown in Table 6.2.

Table 6.2: Time taken for complete segmentation of a range image.

Image Size	Time(sec)
32×32	0.30
64×64	0.92
128×128	2.85
256×256	12.11
512×512	45.37

These numbers reflect the time required for two complete DWT operations in addition to the preprocessing and edge–map to final segmentation operations.

The numbers shown are averages from several images since the edge linking, labeling, and final segmentation modules execution times are highly dependent on the actual image.

CHAPTER 7

Summary and Conclusions

This chapter presents a summary of the work and the contributions which it makes to the areas of range image processing.

7.1 Summary

Analysis of range data is important for performing scene interpretation in robotic environments. In this work, the problem of recovering surfaces from range data was considered. The approach taken used the wavelet transform for the analysis of range data.

In summary, the tasks accomplished in this work are:

- A review of existing techniques for analyzing and segmenting range data.
- A review of wavelet analysis and its applications to signal and image processing.
- A study of the properties of the wavelet transform and their applicability to range image analysis.
- A comparison of the discrete implementations of the wavelet transform and their suitability for range data processing.

- A study of the general properties of wavelets, and a set of conditions necessary for a wavelet family to be used for range image segmentation.
- Two new algorithms that use the transform for the detection of multiscale edges within surfaces in range images.

7.2 Contributions

Two algorithms that use the transform for the detection of multiscale edges within surfaces in range image were developed and evaluated. These algorithms were applied to several synthetic and range images and were found to yield accurate segmentation results in each case. Further, the algorithms were shown to deal successfully with noise present in real range data.

The first algorithm uses the magnitude of the wavelet transform using the derived wavelets to provide a complete boundary representation of the objects inside a range image. This detection of these boundaries is very time–efficient, and provides an input to more elaborate segmentation algorithms, including the second algorithm developed in this work. The role of multiresolution analysis in focusing on features of different sizes within the image was discussed and illustrated using several real range images. These results also demonstrated how the algorithm can be adjusted for different environments that call for multiscale analysis of the objects in them.

The second algorithm performs a complete analysis of the surface orientations in the range image and combines this information with the output from the first image. The fused result contains edges occuring both at the boundary as well as within the objects in the original scene. This edge map is then labeled, giving a final output with different labels for each surface. This image can be used in higher level 3–D reconstruction and recognition processes. This algorithm was applied to several real range images with irregular backgrounds and shown to successfully smooth out noise without losing any of the features in the image. The algorithm was also shown to degrade gracefully with the addition of noise and execute in under three seconds on a Sun SPARCstation 10 for a 128×128 pixels image.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] M. A. Abidi and R. C. Gonzalez (eds.). *Data Fusion in Robotics and Machine Intelligence.* Academic Press, Boston, MA, 1992.

[2] M. Baccar. "Surface Characterization Using a Gaussian Weighted Least Squares Technique Towards Segmentation of Range Images,". Master's thesis, University of Tennessee, Knoxville, 1994.

[3] M. Baccar and M. A. Abidi. "Towards a Reliable Segmentation of Range Images," *SPIE Intl. Symp. on Intelligent Robotics and Computer Vision*, pp. 1024–1036, November 1993.

[4] P. Besl. *Surfaces in Range Image Understanding.* Springer–Verlag, 1988.

[5] P. J. Besl and R. C. Jain. "Invariant Surface Characteristics for 3–D Object Recognition in Range Images," *Computer Vision, Graphics, and Image Processing*, 33:33–80, June 1985.

[6] P. J. Besl and R. C. Jain. "Segmentation Through Variable–Order Surface Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):167–192, March 1988.

[7] P. Bratley and B. L. Fox amd E. L. Schrage. *A Guide to Simulation.* Springer–Verlag, second edition, 1987.

[8] P. J. Burt and E. H. Adelson. "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, 31(4):532–540, April 1983.

[9] A. Calderon. "Intermediate Spaces and Interpolation, the Complex Method," *Studia Math*, 24:113–190, 1964.

[10] J. Canny. "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, June 1986.

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

[12] J. Clark. "Singularity Theory and Phantom Edges in Scale Space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:720–727, July 1988.

[13] D. Colella and C. Heil. "The Characterization of Continuous, Four–Coefficient Scaling Functions and Wavelets," *IEEE Transactions on Information Theory*, 38(2):876–881, March 1992.

[14] T. R. Crimmins. "Geometric Filter for Speckle Reduction," *Applied Optics*, 24(10), May 1985.

[15] I. Daubechies. "Orthonormal Bases of Compactly Supported Wavelets," *Communications in Pure and Applied Mathematics*, 41:909–996, November 1988.

[16] I. Daubechies. "The Wavelet Transform, Time–Frequency Localization and Signal Analysis," *IEEE Transactions on Information Theory*, 36(5):961–1005, September 1990.

[17] I. Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, 1992.

[18] I. Daubechies and J. C. Lagarias. "Two–scale Difference Equations I: Existence and Global Regularity of Solutions," *SIAM Journal of Mathematical Analysis*, 22(5):1388–1410, September 1991.

[19] E. R. Dougherty. *An Introduction to Morphological Image Processing.* SPIE, Bellingham, WA, 1992.

[20] P. Dutilleux. "An Implementation of the 'Algorithme à Trous' to Compute the Wavelet Transform," in J. M. Combes et al, editor, *Proceedings of the International Conference Wavelets: Time–Frequency Methods and Phase Space*, pp. 298–304, Marseille, France, December 1987.

[21] T. J. Fan. *Describing and Recognizing 3–D Objects Using Surface Properties.* Springer–Verlag, New York, NY, 1990.

[22] P. Franklin. "A Set of Continuous Orthogonal Functions," *Math Annal*, 100:522–529, 1928.

[23] J. Froment and S. Mallat. Second generation compact image coding with wavelets. in *Wavelets: A Tutorial in Theory and Applications*, pp. 67–104. Academic Press, 1992.

[24] D. Gabor. "Theory of Communication," *J. IEE (London)*, 93:429–457, 1946.

[25] R. C. Gonzalez, M. Baccar, and M. A. Abidi. "Segmentation of Range Images via Data Fusion and Morphological WaterSheds," in *Proc. of the 8th Scandinavian Conf. on Image Anal.*, pp. 21–35, Norway, May 1993.

[26] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison Wesley, second edition, 1992.

[27] A. Grossmann. Wavelet transform and edge detection. in S. Albeverio et al., editors, *Stochastic Processes in Physics and Engineering.* D. Reidel, 1988.

[28] A. Grossmann and J. Morlet. "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape," *SIAM J. Math. Anal.*, 15:723–736, 1984.

[29] A. Haar. "Zur Theorie der Orthogonalen Funktionensysteme," *Math. Anal*, 69:331–371, 1910.

[30] R. M. Haralick and L. G. Shapiro. "Survey: Image Segmentation Techniques," *Computer Vision, Graphics, and Image Processing*, 29:100–132, 1984.

[31] R.M. Haralick, L.T. Watson, and T.J. Laffey. "The Topographic Primal Sketch," *International Journal on Robotics Res*, 2(1):50–72, 1983.

[32] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. "A Real–Time Algorithm for Signal Analysis with the Help of the Wavelet Transform," in J. M. Combes et al, editor, *Proceedings of the International Conference Wavelets: Time–Frequency Methods and Phase Space*, pp. 286–297, Marseille, France, December 1987.

[33] S. Inokuchi, T. Nita, F. Matsuday, and Y Sakurai. "A Three–Dimensional Edge–Region Operator for Range Pictures," in *Proceedings of the 6th International Conference on Pattern Recognition*, pp. 918–920, Munich, West Germany, October 1982.

[34] J. Kovacevic and M. Vetterli. "Non–Separable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for R^n," *IEEE Transactions on Information Theory: Special Issue on wavelet transforms and multidimensional signal analysis*, January 1992.

[35] R. Kronland-Martinet, J. Morlet, and A. Grossmann. "Analysis of Sound Patterns Through Wavelet Transforms," *International Journal of Pattern Recognition and Artificial Intelligence*, 1(2):273–302, 1987.

[36] A. S. Lewis and G. Knowles. "Image Compression Using the 2–D Wavelet Transform," *IEEE Transactions on Image Processing*, 1:244–250, April 1992.

[37] J. Littlewood and R. Paley. "Theorems on Fourier Series and Power Series," *Proc. London Math. Soc.*, 42:52–89, 1937.

[38] S. Mallat. "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:2091–2110, December 1989.

[39] S. Mallat. "Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(R)$," *Transactions of the American Mathematical Society*, 315(1):69–87, September 1989.

[40] S. Mallat. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:674–693, July 1989.

[41] S. Mallat and W. L. Hwang. "Singularity Detection and Processing with Wavelets," *IEEE Transactions on Information Theory*, 38(2):617–643, March 1992.

[42] S. Mallat and S. Zhong. "Characterization of Signals from Multiscale Edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):710–732, July 1992.

[43] D. C. Marr and E. Hildreth. "Theory of Edge Detection," *Proc. Roy. Soc., London*, B–207:187–217, 1980.

[44] Y. Meyer. *Ondelettes et Operateurs. English title: Wavelets and Operators.* Cambridge University Press, 1992.

[45] A. Mitiche and J. K. Aggarwal. "Detection of Edges Using Range Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):174–178, March 1983.

[46] M. Oshima and Y. Shirai. "Object Recognition Using Three–Dimensional Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4):353–361, July 1983.

[47] B. Parvin and G. Medioni. "Adaptive Multiscale Feature Extraction from Range Data," *Computer Vision, Graphics, and Image Processing*, 45:346–356, 1989.

[48] O. Rioul and P. Duhamel. "Fast Algorithms for Discrete and Continuous Wavelet Transforms," *IEEE Transactions on Information Theory*, 38(2):569–586, March 1992.

[49] O. Rioul and M. Vetterli. "Wavelets and Signal Processing," *IEEE Signal Processing Magazine*, 8:14–38, October 1991.

[50] M. J. Shensa. "The Discrete Wavelet Transform: Wedding the à Trous and Mallat Algorithms," *IEEE Transactions on Signal Processing*, 40(10):2464–2482, October 1992.

[51] J. C. Sluder and M. A. Abidi. "Parallel Range Data Processing: A Real Case Study," *SPIE Intelligent Robots and Computer Vision X*, 1608:248–261, November 1991.

[52] M. J. Smith and T. P. Barnwell. "Exact Reconstruction Techniques for Tree–Structured Subband Coders," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34:434–441, 1986.

[53] G. Strang. "Wavelets and Dilation Equations: A Brief Introduction," *SIAM Review*, 31(4):614–627, December 1989.

[54] L. T. Watson T. C. Pong, L. G. Shapiro and R. M. Haralick. "Experiments in Segmentation Using a Facet Model Region Grower," *Computer Vision, Graphics, and Image Processing*, 25:1–18, 1984.

[55] R. W. Taylor, M. Savini, and A. P. Reeves. "Fast Segmentation of Range Imagery into Planar Regions," *Computer Vision, Graphics, and Image Processing*, 45(1):42–60, January 1989.

[56] N. Yokoa and M. Levine. "Range Image Segmentation Based on Differential Geometry: A Hybrid Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):643–649, June 1989.

VITA

Osama Neiroukh was born in Jerusalem, Israel, in 1970. He graduated from St. George's School in Jerusalem in 1986. He studied at the Rumney College of Technology in Cardiff, Wales, from 1986 to 1988. In 1991 he completed his bachelor's degree in Computer Systems Engineering at the University of Bristol in Bristol, England. In 1992, he entered the Department of Electrical and Computer Engineering's graduate school at the University of Tennessee in Knoxville. He is a member of Phi Kappa Phi and Eta Kappa Nu.